

**POLITECNICO DI MILANO**

**Scuola di Ingegneria dei Sistemi  
Corso di Laurea Magistrale in Ingegneria Matematica**



## **Implementazione su codici a elementi finiti di modelli a grandi vortici per la turbolenza**

Relatore: Dott.ssa Antonella Abbà

Correlatore: Prof. Lorenzo Valdetaro

Tesi di Laurea di:  
Simone Cassani Matr. 725124

**Anno Accademico 2010-2011**

<b>Introduzione</b>	<b>1</b>
<b>1 Modelli a grandi vortici per la turbolenza</b>	<b>4</b>
1.1 Equazioni per la fluidodinamica . . . . .	4
1.2 Teoria e simulazione della turbolenza . . . . .	6
1.3 Simulazione a grandi vortici . . . . .	8
1.3.1 Equazioni di Navier-Stokes filtrate . . . . .	9
1.3.2 Il modello di Smagorinsky . . . . .	10
1.3.3 Procedura di Germano . . . . .	12
<b>2 Formulazione algebrica del problema</b>	<b>15</b>
2.1 Formulazione debole . . . . .	16
2.2 Formulazione di Galerkin . . . . .	18
2.3 Il metodo di Galerkin a elementi finiti . . . . .	19
2.4 Discretizzazione temporale . . . . .	20
2.5 Il termine di sottogriglia . . . . .	23
<b>3 Implementazione in LifeV</b>	<b>25</b>
3.1 Griglia . . . . .	25
3.2 Matrici e Vettori . . . . .	26
3.3 FESpace . . . . .	27
3.4 IFPACK e Aztec00 . . . . .	28
3.5 Il problema di Oseen turbolento . . . . .	28
3.5.1 AssemblyElemental . . . . .	29
3.5.2 OseenSolver . . . . .	34

---

3.5.3	Cylinder . . . . .	37
3.5.4	Il file dei dati in ingresso . . . . .	45
<b>4</b>	<b>Geometrie e parametri del problema</b>	<b>49</b>
4.1	Flusso nel canale cilindrico . . . . .	49
4.1.1	Calcolo della forzante e fattore di conversione . . . . .	51
4.2	Corrente nel canale piano . . . . .	52
<b>5</b>	<b>Risultati: regime laminare</b>	<b>56</b>
5.1	Flusso nel canale cilindrico: $Re_\tau = 90$ . . . . .	56
5.2	Corrente nel canale piano: $Re_\tau = 30$ . . . . .	59
5.2.1	Soluzione analitica . . . . .	61
<b>6</b>	<b>Risultati: regime turbolento</b>	<b>65</b>
6.1	Flusso nel canale cilindrico . . . . .	65
6.1.1	$Re_\tau = 380$ , condizione iniziale casuale . . . . .	67
6.1.2	$Re_\tau = 380$ , condizione iniziale Stokes . . . . .	67
6.1.3	$Re_\tau = 380$ , condizione iniziale logaritmica . . . . .	70
6.2	Corrente nel canale piano . . . . .	78
6.2.1	$Re_\tau = 180$ , disturbo percentuale . . . . .	79
6.2.2	$Re_\tau = 180$ , disturbo omogeneo . . . . .	80
	<b>Conclusioni e sviluppi futuri</b>	<b>86</b>

## Elenco delle figure

1	Fenomeni turbolenti di diverse scale . . . . .	2
1.1	Bande energetiche caratteristiche della turbolenza e domini dei metodi risolutivi. . . . .	8
1.2	Domini risolutivi dei filtri di griglia e test. . . . .	12
2.1	Condizioni naturali superficie di outflow . . . . .	16
2.2	Funzioni di base per il nodo $x_i$ . . . . .	20
2.3	Funzioni di base di tipo bolla e elementi di $\mathcal{P}_{1b}$ . . . . .	21
4.1	Dominio di calcolo, canale cilindrico. . . . .	49
4.2	Canale cilindrico, raffinamento della griglia . . . . .	51
4.3	Dominio di calcolo, canale piano. . . . .	53
4.4	Canale piano, raffinamento della griglia . . . . .	54
5.1	Canale cilindrico $Re_\tau = 90$ . Energia cinetica in funzione del tempo. Confronto fra i diversi modelli di turbolenza. . . . .	57
5.2	Canale cilindrico $Re_\tau = 90$ . Energia cinetica in funzione del tempo. Confronto fra i diversi modelli di turbolenza, dettaglio. . . . .	58
5.3	Canale piano $Re_\tau = 30$ . Energia cinetica in funzione del tempo. Confronto fra i diversi modelli di turbolenza. . . . .	60
5.4	Canale piano $Re_\tau = 15$ . Confronto tra soluzione numerica e analitica. Profili di velocità in direzione $x$ . . . . .	63
6.1	Canale cilindrico $Re_\tau = 380$ . Profili di velocità istantanei in funzione della coordinata $y$ a partire da condizione iniziale casuale. . . . .	68

6.2	Canale cilindrico $Re_\tau = 380$ . Energia cinetica in funzione del tempo a partire da condizione iniziale casuale . . . . .	69
6.3	Canale cilindrico $Re_\tau = 380$ , condizione iniziale Stokes. Soluzione del problema di Stokes stazionario. . . . .	69
6.4	Canale cilindrico $Re_\tau = 380$ , condizione iniziale logaritmica con disturbo al 5%. . . . .	71
6.5	Canale cilindrico $Re_\tau = 380$ . Energia cinetica in funzione del tempo a partire da condizione iniziale logaritmica con disturbo al 5%. . . . .	72
6.6	Canale cilindrico $Re_\tau = 380$ . Profili di velocità istantanei in funzione della coordinata $y$ per diverse ampiezze del filtro $\bar{\Delta}$ a partire da condizione iniziale logaritmica. . . . .	73
6.7	Canale cilindrico $Re_\tau = 380$ . Velocità istantanee sul piano $yz$ al tempo $t = 1.48$ , $x = 0$ . . . . .	75
6.8	Canale cilindrico $Re_\tau = 380$ . Velocità medie istantanee in funzione della distanza dal bordo per differenti istanti temporali, a partire da condizione iniziale logaritmica con disturbo $o(1)$ . . . . .	76
6.9	Canale cilindrico $Re_\tau = 380$ . Fluttuazioni di velocità istantanee in funzione della distanza dal bordo per differenti istanti temporali, a partire da condizione iniziale logaritmica con disturbo $o(1)$ . . . . .	77
6.10	Canale piano $Re_\tau = 180$ . Energia cinetica in funzione del tempo, a partire da condizione iniziale logaritmica con disturbo percentuale di diverse intensità. . . . .	79
6.11	Canale piano $Re_\tau = 180$ . Profili di velocità istantanei in funzione della coordinata $z$ per diverse ampiezze del filtro $\bar{\Delta}$ a partite da condizione iniziale logaritmica con disturbo percentuale al 5%. . . . .	81
6.12	Canale piano $Re_\tau = 180$ . Profili di velocità istantanei in funzione della coordinata $z$ , a partire da condizione iniziale logaritmica con disturbo omogeneo $o(1)$ . . . . .	83
6.13	Canale piano $Re_\tau = 180$ . Profili di velocità istantanei medi in funzione della distanza dal bordo $z$ a partire da condizione iniziale logaritmica con disturbo omogeneo $o(2)$ , $t = 1$ . . . . .	84
6.14	Canale piano $Re_\tau = 180$ . Fluttuazioni di velocità istantanee in funzione della distanza dal bordo $z$ a partire da condizione iniziale logaritmica con disturbo omogeneo $o(2)$ , $t = 1$ . . . . .	85

## Elenco delle tabelle

1.1	Filtri monodimensionali e relative trasformate di Fourier . . . . .	9
3.1	Tempi medi per iterazione. . . . .	47
6.1	Regime turbolento, flusso nel canale cilindrico, $Re_{\tau} = 380$ : parametri e risultati delle simulazioni. . . . .	66
6.2	Regime turbolento, corrente nel canale piano. $Re_{\tau} = 180$ : parametri e risultati delle simulazioni . . . . .	78

## Riassunto della tesi

Questo lavoro di tesi Magistrale in Ingegneria Matematica si pone nell'ambito della modellazione numerica agli elementi finiti di equazioni a derivate parziali per la fluidodinamica. La turbolenza riveste un ruolo importante in questo settore poiché molti dei fenomeni fisici studiati ne sono caratterizzati. A questo scopo il modello di turbolenza per la simulazione ai grandi vortici, LES, è stato integrato nella libreria agli elementi finiti LifeV.

La tesi è divisa in sei capitoli.

Nel capitolo 1 vengono descritte le equazioni di Navier-Stokes e si introduce il modello sottogriglia di Smagorinsky per la simulazione ai grandi vortici.

Il capitolo 2 tratta la modellazione numerica delle equazioni Navier-Stokes e l'introduzione al metodo agli elementi finiti.

Nel capitolo 3 è mostrata l'implementazione del problema nella libreria LifeV e vengono descritte le nuove funzioni introdotte.

Nel capitolo 4 è presente la descrizione delle geometrie dei problemi affrontati e il calcolo dei parametri utilizzati.

Nel capitolo 5 sono analizzati i risultati ottenuti in regime laminare.

Il capitolo 6 tratta l'analisi dei risultati ottenuti in regime turbolento.

## Abstract

This Master thesis in Ingegneria Matematica deals with numerical finite elements modeling of fluid dynamics differential equations. Turbulence plays an important role in this area since characterizes many physical phenomena. The aim of this work is the implementation of turbulence models for large eddy simulation, LES, in the finite element library LifeV.

The thesis is composed by six chapters.

Chapter 1 is devoted to the description of Navier-Stokes equations and the introduction to Smagorinsky subgrid model for large eddy simulation.

Chapter 2 deals with numerical modeling of Navier-Stokes equations and introduction to finite element method.

Chapter 3 shows the implementation of the model in LifeV library and the description of new features.

Chapter 4 gives a description of covered problems geometry and data computation for numerical simulations.

Chapter 5 discusses laminar flow results.

Chapter 6 discusses turbulent flow results.



## Introduzione

La turbolenza nei fenomeni fluidodinamici è la regola, non l'eccezione. La maggior parte dei fenomeni che accadono nel quotidiano sono caratterizzati infatti da flussi turbolenti.

Lo strato limite dell'atmosfera terrestre, le correnti oceaniche e gli uragani sono manifestazioni turbolente su larga scala. Ma anche molti fenomeni su scale medie come il flusso d'aria intorno al profilo alare di un aereo e le correnti nei fiumi sono turbolenti. Il regime turbolento è presente anche in fenomeni di dimensioni caratteristiche minori come il lancio di una pallina da golf e il fumo di sigaretta.

La turbolenza riguarda anche molti settori industriali tra cui quello aerospaziale e automobilistico per problemi relativi all'aerodinamica e ai flussi nei condotti dei motori e il settore chimico che la sfrutta per esempio nel mescolamento di materiali in fase gassosa.

Le strutture caratteristiche della turbolenza sono vortici, di grandi, medie e piccole dimensioni, che interagiscono tra loro. Il campo di velocità presenta delle fluttuazioni aleatorie spaziali e temporali, che contengono un ampio spettro di frequenze. Un'altra caratteristica comune ai flussi turbolenti è la sensibilità alle condizioni iniziali. Per questo motivo il comportamento di questi flussi è difficilmente prevedibile su tempi lunghi.

Lo studio della turbolenza ha origini che risalgono al XV secolo con Leonardo da Vinci, anche se la svolta decisiva avvenne verso la fine del XIX secolo, quando Reynolds ne individuò i regimi caratteristici. L'evoluzione tecnologica permise negli anni 30' del XX secolo di ottenere misure sulla turbolenza sempre più accurate. Grazie a queste migliorie Taylor formulò la teoria statistica della turbolenza isotropa, mentre Kolmogorov qualche anno più tardi formulò la teoria statistica della turbo-



(a) uragano



(b) vortici turbolenti generati da un aereo



(c) fumo di sigaretta

**Figura 1:** Fenomeni turbolenti di diverse scale

lenza localmente isotropa che poteva essere applicata a flussi turbolenti reali.

Il progresso scientifico ha reso possibile il rilevamento di dati sempre più precisi e accurati. Il passo più importante avvenne negli anni 70' del XX secolo con l'avvento dei calcolatori e quindi della possibilità, che diventerà concreta nel decennio successivo, della simulazione numerica turbolenta. Uno dei modelli per la turbolenza sviluppato in questo periodo è il modello *LES* per la simulazione ai grandi vortici [1] [9] [17]. La simulazione numerica fluidodinamica e la simulazione della turbolenza sono quindi strettamente correlate.

La simulazione numerica a grandi vortici è ampiamente utilizzata con ottimi risultati su griglie strutturate. In questo lavoro vengono effettuate simulazioni su griglie non strutturate, argomento che è tuttora oggetto di studi. La calibrazione del modello mediante una ricerca dei parametri ottimali, diventa quindi un passaggio obbligatorio per ottenere risultati coerenti con la fisica del problema.

In questo contesto si inserisce il lavoro proposto che ha come scopo l'introduzione della tecnica *LES* nella libreria per la simulazione numerica agli elementi finiti *LifeV*. *LifeV* è un progetto iniziato nel 1999/2000 dalla collaborazione del Politecnico di Milano (MOX), École Polytechnique Fédérale de Lausanne (CMCS) e INRIA (Progetto REO).

*LifeV* è una libreria a elementi finiti scritta in linguaggio C++ per la risoluzione di equazioni a derivate parziali.

I campi applicativi di questa libreria sono molteplici: problemi di dinamica dei fluidi e delle strutture, trasporto di calore, interazioni fluido-struttura, trasporto in mezzi porosi e simulazioni multiscala. Il codice è stato inoltre impiegato con successo in campo industriale.

Un notevole passo avanti nell'organizzazione e nella documentazione del codice si è avuto nel dicembre 2010 durante le *cleaning weeks*, settimane in cui la libreria e la relativa documentazione sono state controllate dal team di sviluppatori di *LifeV*.

## Modelli a grandi vortici per la turbolenza

### 1.1 Equazioni per la fluidodinamica

Le equazioni che descrivono il moto di un fluido, scritte in forma conservativa sono

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ \frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot \mathbf{u} + p \overleftrightarrow{\mathbf{I}} - \lambda \nabla \cdot \mathbf{u} \overleftrightarrow{\mathbf{I}} - 2\mu \overleftrightarrow{\mathbf{S}}) &= \rho \mathbf{f}, \\ \frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho h \mathbf{u} - \lambda \mathbf{u} \nabla \cdot \mathbf{u} - 2\mu \overleftrightarrow{\mathbf{S}} \cdot \mathbf{u} - \chi \nabla \theta) &= \rho(\mathbf{f} \cdot \mathbf{u} + r). \end{aligned} \quad (1.1)$$

dove  $\rho$  è la densità del fluido,  $\mathbf{u}$  la sua velocità,  $p$  la pressione,  $\lambda$  e  $\mu$  sono rispettivamente il secondo coefficiente di viscosità e il coefficiente di viscosità dinamico.  $\overleftrightarrow{\mathbf{I}}$  è il tensore identità, mentre  $\overleftrightarrow{\mathbf{S}} = \frac{1}{2}(\nabla \mathbf{u} + \nabla^T \mathbf{u})$  è il tensore velocità di deformazione e  $\mathbf{f}$  la forza specifica esterna.  $E$  è l'energia totale,  $h = E + \frac{p}{\rho}$  l'entalpia specifica totale,  $\chi$  il coefficiente di conduttività termica dinamica,  $\theta$  la temperatura e  $r$  è un termine di produzione di energia interna per unità di massa.

La prima equazione di (1.1) deriva dalla applicazione della conservazione della massa all'interno di un volume di riferimento. La seconda, che è in forma vettoriale, si ottiene dall'equazione della quantità di moto, mentre l'ultima è l'equazione di conservazione dell'energia. La terza equazione di (1.1) richiede inoltre l'utilizzo di relazioni termodinamiche per modellare alcuni dei termini che vi compaiono e chiudere il sistema.

Sotto l'ipotesi di fluido incomprimibile si può ottenere un sistema semplificato che ne descrive il moto. In questo caso infatti l'ultima equazione di (1.1) viene sostituita da

$$\nabla \cdot \mathbf{u} = 0. \quad (1.2)$$

La (1.2) è infatti sufficiente a chiudere il sistema.

Inoltre se il fluido è a densità uniforme la prima equazione di (1.1) non è più necessaria.

Nel caso di un fluido incomprimibile a proprietà costanti, è possibile ottenere la formulazione

$$\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot \mathbf{u}) &= -\nabla \cdot (\mathbf{p} \overleftrightarrow{\mathbf{I}} - 2\mu \overleftrightarrow{\mathbf{S}}), \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \quad (1.3)$$

Attraverso una procedura di adimensionalizzazione e alcuni passaggi matematici si ottengono le equazioni di *Navier-Stokes* incomprimibili a viscosità e densità costanti

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla(p) - \frac{1}{\text{Re}} \Delta \mathbf{u} &= 0, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \quad (1.4)$$

Il parametro  $\text{Re}$ :

$$\text{Re} = \frac{\mathbf{U}L}{\nu}, \quad (1.5)$$

è detto *numero di Reynolds*.  $\mathbf{U}$  è la velocità media del fluido,  $\nu = \frac{\mu}{\rho}$  la viscosità cinematica e  $L$  la lunghezza caratteristica del problema<sup>1</sup>.

Il numero di Reynolds esprime il rapporto tra forze inerziali, termine al numeratore, e forze viscosi, termine al denominatore, e fornisce informazioni sul moto del fluido. È infatti possibile determinare per ogni problema un parametro  $\text{Re}_c$  detto numero di Reynolds critico. Per valori di  $\text{Re}$  inferiori a  $\text{Re}_c$  il fluido si troverà in regime *laminare*, mentre per valori di  $\text{Re}$  superiori a  $\text{Re}_c$  il fluido si troverà in regime *turbolento*. Mentre il regime laminare è caratterizzato da un moto che avviene per scorrimento di strati infinitesimi, senza rimescolamento, il regime turbolento è caratterizzato da forze inerziali, molto maggiori rispetto alle componenti viscosi, che impongono forti gradienti di velocità. Si ottiene così un moto non ordinato caratterizzato da vortici interni. Il regime turbolento è caratterizzato da un moto deterministico, la cui evoluzione è determinata univocamente dallo stato attuale del sistema, anche se l'instabilità che lo caratterizza fa sì che piccole perturbazioni sui dati possano generare variazioni importanti sull'evoluzione del flusso.

Analizzando il moto di un fluido in un canale in regime turbolento a  $\text{Re}$  elevato è possibile notare come il termine convettivo sia predominante nella parte centrale del canale, mentre vicino alle pareti ci sia una predominanza del termine viscoso.

<sup>1</sup>La derivazione rigorosa delle equazioni (1.1) e i passaggi relativi alla loro adimensionalizzazione sono disponibili in [19]

Infatti le condizioni di non scivolamento delle pareti portano alla formazione di uno *strato limite viscoso*, una regione in cui il moto del fluido è caratterizzato da regime laminare. Al confine di questa regione termini convettivi e viscosi hanno lo stesso ordine di grandezza. Dal confronto tra i due rispettivi contributi:

$$\text{termine convettivo : } \mathbf{u} \cdot \nabla \mathbf{u} \simeq \frac{U^2}{L},$$

$$\text{termine viscoso : } \nu \Delta \mathbf{u} \simeq \nu \frac{U}{\delta^2},$$

si ottiene:

$$\mathbf{u} \cdot \nabla \mathbf{u} \simeq \nu \Delta \mathbf{u} \Rightarrow \delta \simeq \frac{L}{\sqrt{Re}},$$

dove  $\delta$  è lo spessore dello strato limite.

La formulazione (1.4) non è l'unica possibile formulazione adimensionale per le equazioni di Navier-Stokes. È infatti possibile utilizzare un numero di Reynolds basato sulla velocità d'attrito a parete

$$V_* = \sqrt{\frac{\tau_w}{\rho}}, \quad (1.6)$$

dove  $\tau_w$  è lo sforzo a parete.

Il risultato di queste operazioni è:

$$Re_\tau = \frac{V_* L}{\nu}. \quad (1.7)$$

Con questa notazione ci si trova nello strato limite per valori di  $y^+ < 5$ , dove

$$y^+ = Re_\tau |y - y_w|. \quad (1.8)$$

$y_w$  è il valore della coordinata a parete.

## 1.2 Teoria e simulazione della turbolenza

Esistono diverse tecniche per la simulazione del regime turbolento, il cui utilizzo è in funzione delle caratteristiche del moto cui si è principalmente interessati e della precisione desiderata.

La prima tecnica descritta è la Simulazione Numerica Diretta (DNS)<sup>2</sup> delle equazioni di Navier Stokes.

La DNS è in grado di rappresentare tutte le strutture turbolente presenti nella corrente, descrivendo la dinamica dei vortici fino alle scale di Kolmogorov. Per fare ciò

<sup>2</sup>DNS: Direct Numerical Simulation

necessita di una griglia con passo minore delle più piccole scale dissipative presenti nel moto.

Questa tecnica necessita di schemi numerici accurati e di ordine elevato al fine di contenere gli errori. Questi schemi sono caratterizzati da un alto costo computazionale, che ne limita l'utilizzo a correnti semplici con numeri di Reynolds non elevati. Queste caratteristiche rappresentano il limite maggiore della DNS.

Una tecnica alternativa si ottiene applicando la media d'insieme alle (1.4). Si ottiene così la tecnica delle equazioni di Navier-Stokes mediate alla Reynolds (RANS)<sup>3</sup>. La procedura utilizzata nella tecnica RANS permette di ottenere indicazioni solamente sull'evoluzione delle quantità medie del moto. Necessita inoltre di modelli di turbolenza per la chiusura del sistema. Risulta essere quindi un metodo meno oneroso in termini computazionali, ma limitato nella qualità delle informazioni fornite. Il modello RANS non è in grado di descrivere dettagliatamente le strutture vorticose e tutto ciò che non fa parte del moto medio del sistema. Inoltre i modelli introdotti per la chiusura del sistema, dovendo risultare specifici per il singolo problema, sono poco universali.

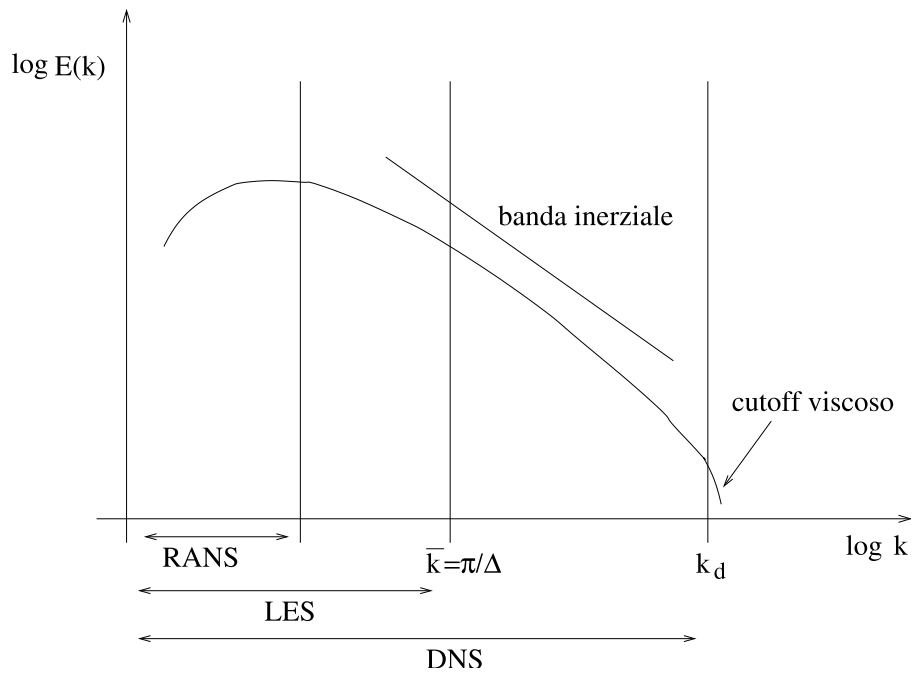
La tecnica utilizzata per la simulazione del moto turbolento in questo lavoro è la Simulazione a Grandi Vortici (LES)<sup>4</sup>. Questa tecnica utilizza un approccio intermedio tra le due presentate in precedenza DNS e RANS.

La LES risolve esattamente le scale grandi e medie della turbolenza, caratterizzate da basse frequenze, mentre modella le piccole scale, caratterizzate da alte frequenze. La teoria di Kolmogorov [12] evidenzia come le scale più piccole della turbolenza abbiano caratteristiche universali, non dipendenti dalle condizioni al contorno. Questo risultato permette di considerare valida l'approssimazione fornita dalla tecnica LES, il cui modello completo per la turbolenza risulta semplificato e più universale. La tecnica LES si dimostra quindi più accurata della tecnica RANS, nella simulazione delle piccole scale, ma soprattutto meno costosa dal punto di vista computazionale della tecnica DNS. Questo permette simulazioni a numeri di Reynolds più elevati e di trattare quindi casi più realistici.

La Figura 1.1 riassume i diversi livelli di risoluzione dei metodi citati.

<sup>3</sup>RANS: Reynolds Averaged Navier-Stokes

<sup>4</sup>LES: Large Eddy Simulation



**Figura 1.1:** Bande energetiche caratteristiche della turbolenza e domini dei metodi risolutivi.

### 1.3 Simulazione a grandi vortici

L'operazione che caratterizza il metodo LES e permette la separazione delle scale risolte numericamente, scale a dimensioni maggiori, da quelle modellate, scale a dimensioni minori, è il filtraggio. Questo si traduce in termini matematici in un integrale di convoluzione tra la funzione che si vuole filtrare  $f$  e la funzione filtro normalizzata  $G_{\bar{\Delta}}$ . Il parametro  $\bar{\Delta}$  rappresenta l'ampiezza del filtro e di conseguenza la dimensione considerata per le piccole scale. L'espressione che si ottiene per la variabile filtrata è

$$\bar{f}(\mathbf{x}, t) = \int_{\Omega} f(\mathbf{x} - \mathbf{r}, t) G(\mathbf{r}, \mathbf{x}) d^3 \mathbf{r}, \quad (1.9)$$

dove

$$\int_{\Omega} G(\mathbf{r}, \mathbf{x}) d^3 \mathbf{r} = 1.$$

Il filtraggio produce la variabile residua

$$f'(\mathbf{x}, t) = f(\mathbf{x}, t) - \bar{f}(\mathbf{x}, t),$$

che verrà calcolata a partire dalle scale risolte e rappresenta il contributo dei vortici turbolenti più piccoli. Si ha in generale

$$\overline{f'(\mathbf{x}, t)} \neq 0.$$



Nome	funzione filtro	funzione di trasferimento
Generale	$G(\mathbf{r})$	$\widehat{G}(\mathbf{k}) = \int_{-\infty}^{+\infty} e^{i\mathbf{k}\mathbf{r}} G(\mathbf{r}) d\mathbf{r}$
Box	$\frac{1}{\Delta} H(\frac{1}{2}\Delta -  \mathbf{r} )$	$\frac{\sin(\frac{1}{2}\mathbf{k}\Delta)}{\frac{1}{2}\mathbf{k}\Delta}$
Gaussiano	$\frac{6}{\pi\Delta^2} e^{-6\mathbf{r}^2/\Delta^2}$	$e^{-\mathbf{k}^2\Delta^2/24}$
Sharp cutoff	$\frac{\sin(\pi\mathbf{r}/\Delta)}{\pi\mathbf{r}}$	$H(\frac{\pi}{\Delta} -  \mathbf{k} )$

Tabella 1.1: Filtri monodimensionali e relative trasformate di Fourier.  $H$  è la funzione di Heavyside.

La funzione filtro viene detta omogenea se non dipende dalla coordinata  $\mathbf{x}$  e isotropa se la dipendenza da  $\mathbf{r}$  è ridotta solamente a  $|\mathbf{r}|$ .

Nel caso di equazioni a derivate parziali, è necessario individuare il comportamento dell'operazione di filtraggio rispetto alla derivazione. Per quanto riguarda la derivata temporale, filtraggio e derivazione commutano, si ha quindi

$$\frac{\partial \bar{f}}{\partial t} = \overline{\frac{\partial f}{\partial t}}$$

La situazione è differente invece nel caso di derivata spaziale. Si ottiene infatti un errore di commutazione pari a

$$\frac{\partial \bar{f}}{\partial x_i} - \overline{\frac{\partial f}{\partial x_i}} = \int_{\Omega} f(\mathbf{x} - \mathbf{r}, t) \frac{\partial G(\mathbf{r}, \mathbf{x})}{\partial x_i} d\mathbf{r},$$

che si azzera nel solo caso di filtro omogeneo. La Tabella 1.1 mostra le funzioni di trasferimento per alcuni filtri omogenei monodimensionali.

I filtri mostrati possono essere facilmente estesi al caso di 2 o 3 dimensioni.

### 1.3.1 Equazioni di Navier-Stokes filtrate

La (1.3) si può esprimere in notazione indiciale come segue

$$\begin{aligned} \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} (u_i u_j) &= -\frac{\partial \tilde{p}}{\partial x_i} + 2\nu \frac{\partial S_{ij}}{\partial x_j}, \\ \frac{\partial u_i}{\partial x_i} &= 0, \end{aligned} \quad (1.10)$$

dove tutti i termini sono stati divisi per la densità  $\rho$ . La pressione  $\tilde{p} = \frac{p}{\rho}$ , verrà in seguito indicata come  $p$  per non appesantire la notazione.

La notazione indiciale per il tensore velocità di deformazione è:

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

Applicando il filtro alla seconda equazione di (1.10), sotto ipotesi di filtro omogeneo, sfruttando la commutazione tra filtro e derivata, si ottiene

$$\frac{\partial \bar{u}_i}{\partial x_i} = \frac{\partial u_i}{\partial x_i} = 0, \quad \frac{\partial u_i'}{\partial x_i} = \frac{\partial (u_i - \bar{u}_i)}{\partial x_i} = 0.$$

Applicando poi il medesimo filtro alla equazione di conservazione della quantità di moto, (1.10) prima equazione, si ottengono le equazioni di Navier-Stokes filtrate per un fluido incomprimibile a proprietà costanti:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_i \bar{u}_j) = -\frac{\partial \bar{p}}{\partial x_i} + 2\nu \frac{\partial \bar{S}_{ij}}{\partial x_j}. \quad (1.11)$$

Il termine  $\frac{\partial}{\partial x_j} (\bar{u}_i \bar{u}_j)$  non può però essere calcolato a partire dalle singole componenti. Si introduce quindi il *tensore degli sforzi di sottogriglia*

$$\tau_{ij} = \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j.$$

Si ottiene in questo modo il modello LES per le equazioni di Navier-Stokes

$$\begin{aligned} \frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_i \bar{u}_j) &= -\frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + 2\nu \frac{\partial \bar{S}_{ij}}{\partial x_j}, \\ \frac{\partial \bar{u}_i}{\partial x_i} &= 0. \end{aligned} \quad (1.12)$$

La modellazione del termine di sottogriglia verrà affrontata nei paragrafi successivi.

### 1.3.2 Il modello di Smagorinsky

Questo modello fa parte di quelli che in letteratura sono definiti come *modelli a viscosità turbolenta*. I modelli a viscosità turbolenta sono basati sull'ipotesi che le piccole scale dissipano tutta l'energia che ricevono dalle grandi scale. Modellano quindi il contributo dissipativo delle piccole scale  $\tau_{ij}$  aggiungendo al problema un termine viscoso proporzionale al tensore velocità di deformazione filtrato

$$\tau_{ij}^a = \tau_{ij} - \frac{\delta_{ij}}{3} \tau_{kk} = -2\nu_T \bar{S}_{ij}.$$

Il parametro di proporzionalità  $\nu_T$  è il coefficiente di viscosità turbolenta.

Il modello di Smagorinsky considera la dipendenza di  $\nu_T$  dal solo tasso di dissipazione  $\epsilon$  e dall'ampiezza  $\bar{\Delta}$  del filtro utilizzato. Si ottiene attraverso considerazioni dimensionali l'espressione

$$\nu_T = C\epsilon^{1/3}\bar{\Delta}^{4/3}. \quad (1.13)$$

Se si considera di operare nella banda inerziale, utilizzando quindi valori di  $\bar{\Delta}$  dell'ordine delle sue dimensioni caratteristiche, è possibile trascurare la dissipazione di energia dovuta agli effetti viscosi. Grazie a questa considerazione è possibile esprimere  $\epsilon$  come

$$\epsilon = -\tau_{ij}^a \bar{S}_{ij} = 2\nu_T \bar{S}_{ij} \bar{S}_{ij} = \nu_T |\bar{S}|^2,$$

dove  $|\bar{S}|^2 = 2\bar{S}_{ij}\bar{S}_{ij}$ . La (1.13) diventa quindi

$$\epsilon^{2/3} = C\bar{\Delta}^{4/3} |\bar{S}|^2.$$

Posto  $C_S = C^{3/4}$  si ottiene:

$$\nu_T = C_S^2 \bar{\Delta}^2 |\bar{S}|, \quad \tau_{ij}^a = -2C_S^2 \bar{\Delta}^2 |\bar{S}| \bar{S}_{ij}. \quad (1.14)$$

Le (1.14) mostrano la forma di  $\nu_T$  e  $\tau_{ij}$  che verranno utilizzate nel corso delle prime simulazioni numeriche per la turbolenza, utilizzando il metodo LES.

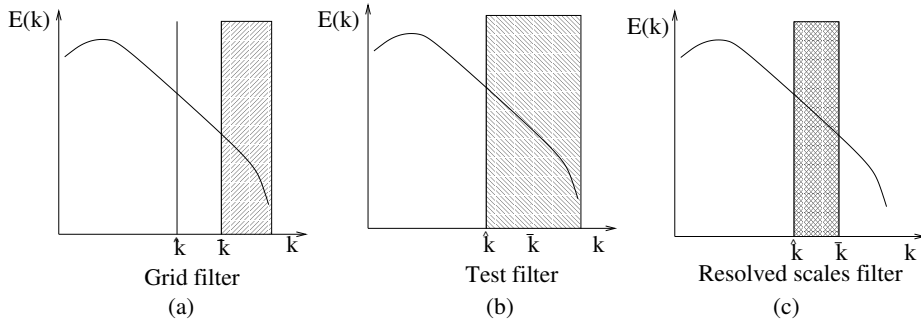
Il modello di Smagorinsky presenta tuttavia alcuni limiti e svantaggi:

- Il valore della costante  $C_S$  deve essere fornito a priori. Esistono in letteratura dei valori di riferimento per tale costante che variano da  $C_S = 0.1 \div 0.23$  a seconda del problema considerato.
- Il modello necessita una correzione nel caso di flusso laminare con grandi gradienti di velocità, per evitare l'introduzione di una eccessiva dissipazione. Per la correzione nello strato limite si utilizza il modello di Van Driest della viscosità turbolenta:

$$\nu_T = \left[ C_S \bar{\Delta} (1 - e^{-y^+/A}) \right]^2 |\bar{S}|, \quad (1.15)$$

dove  $y^+ = \frac{V_* y}{\nu}$  e  $A = 25$  è una costante del metodo.

- Il flusso a piccola scala viene supposto statisticamente omogeneo e isotropo e si assume valida la teoria di Kolmogorov.
- Il modello non considera il *backscatter*, il trasferimento di energia dalle scale più piccole a quelle più grandi, riscontrato nelle simulazioni DNS.



**Figura 1.2:** Domini risolutivi dei filtri di griglia e test.

- L'equazione per  $\nu_T$  forza i tensori  $\tau_{ij}$  e  $S_{ij}$  ad essere paralleli, fatto in contrasto con i risultati di simulazioni DNS.

Il modello di Smagorinsky può essere raffinato e alcuni svantaggi eliminati attraverso la procedura di Germano.

### 1.3.3 Procedura di Germano

La procedura dinamica di Germano [5] non considera più il parametro  $C_S$  come una costante del modello, ma lo calcola nel corso della simulazione in funzione delle coordinate spaziali. A questo proposito viene introdotto, in aggiunta al filtro di griglia di ampiezza  $\bar{\Delta}$ , un secondo filtro, chiamato filtro test, di ampiezza  $\hat{\Delta} > \bar{\Delta}$ . Le ampiezze di entrambi i filtri sono tali per cui essi appartengono alla banda inerziale e il modello filtrato viene supposto valido anche per il filtro test, Figura 1.2.

Per illustrarne il funzionamento si applica di seguito la procedura dinamica all'equazione differenziale

$$\frac{\partial \mathbf{u}(x, t)}{\partial t} = f(\mathbf{u}(x, t)),$$

dove  $f$  è un operatore non lineare.

Applicando il filtro di ampiezza  $\bar{\Delta}$  si ottiene

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} = \overline{f(\mathbf{u})} = f(\bar{\mathbf{u}}) + (\overline{f(\mathbf{u})} - f(\bar{\mathbf{u}})).$$

Supponendo di poter modellare  $\overline{f(\mathbf{u})} - f(\bar{\mathbf{u}})$  con una funzione  $g(\mathbf{c}, \bar{\mathbf{u}}, \bar{\Delta})$ , dove  $\mathbf{c} = (c_1, \dots, c_n)$  è un vettore di costanti indipendenti da  $\bar{\Delta}$  e applicando la stessa procedura con il filtro test si ottiene:

$$\overline{f(\mathbf{u})} - f(\bar{\mathbf{u}}) = g(\mathbf{c}, \bar{\mathbf{u}}, \bar{\Delta}), \quad (1.16)$$

$$\widehat{f(\mathbf{u})} - f(\hat{\mathbf{u}}) = g(\mathbf{c}, \hat{\mathbf{u}}, \hat{\Delta}). \quad (1.17)$$

Applicando i due filtri in successione si arriva all'equazione

$$\widehat{f(\mathbf{u})} - f(\widehat{\mathbf{u}}) = \widehat{g}(\mathbf{c}, \bar{\mathbf{u}}, \bar{\Delta}).$$

Sotto le ipotesi

$$\widehat{f(\mathbf{u})} = f(\widehat{\mathbf{u}}), \quad \widehat{\mathbf{u}} = \widehat{\mathbf{u}},$$

sottraendo la (1.16) alla (1.17) si ottiene:

$$f(\widehat{\mathbf{u}}) - f(\widehat{\mathbf{u}}) = g(\mathbf{c}, \widehat{\mathbf{u}}, \widehat{\Delta}) - \widehat{g}(\mathbf{c}, \bar{\mathbf{u}}, \bar{\Delta}). \quad (1.18)$$

La risoluzione di (1.18) in ogni nodo della griglia di calcolo permette di calcolare i valori delle componenti del vettore  $\mathbf{c}$ .

Si esegue ora la procedura dinamica di Germano sulle equazioni di Navier-Stokes a cui è stato applicato il modello di Smagorinsky.

Si considera il tensore degli sforzi per il filtro di griglia

$$\tau_{ij} = \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j,$$

e il suo corrispondente per il filtro test

$$T_{ij} = \widehat{u}_i \widehat{u}_j - \widehat{u}_i \widehat{u}_j.$$

Si definisce inoltre il tensore degli sforzi risolti

$$L_{ij} = \widehat{u}_i \widehat{u}_j - \widehat{u}_i \widehat{u}_j.$$

È ora possibile riscrivere l'equazione (1.18) per il problema considerato:

$$L_{ij} = T_{ij} - \widehat{\tau}_{ij}. \quad (1.19)$$

L'equazione (1.19) prende il nome di identità di Germano.

Le quantità caratteristiche del modello di Smagorinsky sono

$$\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2C\bar{\Delta}^2 |\bar{S}| \bar{S}_{ij}, \quad \bar{S}_{ij} = \frac{1}{2} \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right),$$

$$T_{ij} - \frac{1}{3} T_{kk} \delta_{ij} = -2C\widehat{\Delta}^2 |\widehat{S}| \widehat{S}_{ij}, \quad \widehat{S}_{ij} = \frac{1}{2} \left( \frac{\partial \widehat{u}_i}{\partial x_j} + \frac{\partial \widehat{u}_j}{\partial x_i} \right),$$

dalle quali si ottiene, applicando l'identità di Germano (1.19),

$$L_{ij}^a \equiv L_{ij} - \frac{1}{3} L_{kk} \delta_{ij} = -2 \left[ C\widehat{\Delta}^2 |\widehat{S}| \widehat{S}_{ij} - C\bar{\Delta}^2 |\bar{S}| \bar{S}_{ij} \right]. \quad (1.20)$$

Il termine a sinistra dell'uguale può essere calcolato esplicitamente e ciò permette la risoluzione del sistema (1.20).

Il sistema (1.20) è composto da 5 equazioni (i tensori che vi compaiono sono simmetrici a traccia nulla) per la sola incognita  $C$ , la sua soluzione si ottiene quindi attraverso una procedura ai minimi quadrati [14].

Definite

$$Q = (L_{ij}^a - 2CM_{ij})^2, \quad M_{ij} = \overline{\Delta^2 |\widehat{S}| \widehat{S}_{ij}} - \widehat{\Delta^2 |\widehat{S}| \widehat{S}_{ij}},$$

la soluzione ottimale del sistema si ottiene per  $\frac{\partial Q}{\partial C} = 0$  che equivale a

$$C(\mathbf{x}, t) = \frac{L_{ij}^a M_{ij}}{2M_{kl} M_{kl}}. \quad (1.21)$$

Dopo averne dato una descrizione è ora possibile riassumere le principali proprietà del modello dinamico di Germano:

- La principale caratteristica che rende il modello di Germano più accurato di quello di Smagorinsky è che la costante  $C$  non è più un dato a priori del problema, ma viene calcolata in funzione dello spazio e del tempo nel corso della simulazione. In questo modo la viscosità turbolenta viene aggiunta localmente in maniera ottimale.
- Il modello è in grado di annullarsi nelle regioni in regime laminare, come ad esempio lo strato limite. In queste zone, non essendo presenti le piccole scale del moto turbolento,  $\overline{S}$  e  $\widehat{S}$  e le velocità  $\overline{u}$  e  $\widehat{u}$  tendono a diventare uguali. Il risultato è che la quantità  $L_{ij}^a$  tende a zero mentre  $M_{ij}$  non tende a zero. Inserendo questi particolari valori nella formula (1.21) si nota come il valore di  $C$  in queste regioni tenda ad annullarsi spegnendo così il modello.
- Ammette anche valori negativi per la costante  $C$ . Valori di  $C < 0$  sono infatti responsabili del fenomeno del *backscatter* o cascata inversa di energia.
- Il modello è soggetto a instabilità numeriche nel caso di valori troppo grandi di  $C$  o di una sua elevata variabilità nello spazio. Per questo solitamente si pongono un limite superiore e inferiore per i valori della costante  $C$ .
- Il modello di Germano, come quello di Smagorinsky, forza il tensore degli sforzi di sottogriglia ad essere allineato con il tensore velocità di deformazione.

## Formulazione algebrica del problema

Si consideri il problema di Navier-Stokes nel dominio  $\Omega \subset \mathbb{R}^d$

$$\left\{ \begin{array}{ll} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) - \nabla \cdot (2\mu \mathbf{S}(\mathbf{u}) - p\mathbf{I}) = \mathbf{f} & \text{in } (0, T) \times \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } (0, T) \times \Omega, \\ \mathbf{u} = \mathbf{g} & \text{su } \Gamma_D, \\ 2\mu \mathbf{S} - p\mathbf{I} = \eta & \text{su } \Gamma_N = \partial\Omega \setminus \Gamma_D. \end{array} \right. \quad (2.1)$$

$\Gamma_D$  e  $\Gamma_N$  rappresentano rispettivamente la frontiera di Dirichlet e Neumann.

Nel caso bidimensionale il problema (2.1) è ben posto. Le soluzioni trovate hanno derivate continue e non mostrano singolarità nella loro evoluzione temporale per dati iniziali regolari.

Nel caso tridimensionale esistenza e unicità della soluzione sono dimostrate solo per tempi piccoli. Si introduce quindi la formulazione debole del problema, per cui è possibile dimostrare un risultato di esistenza globale. Resta ancora aperto invece il problema dell'unicità della soluzione.

Le condizioni al bordo utilizzate sul bordo  $\Gamma_N$ , sono condizioni *naturali* di sforzo normale nullo

$$(2\mu \mathbf{S}(\mathbf{u}) - p\mathbf{I}) \cdot \mathbf{n} = 0. \quad (2.2)$$

Questo particolare tipo di condizioni al contorno non ha significato fisico, ma deriva direttamente dalla formulazione debole.

Se imposta come condizione di outflow per flussi in un canale, la condizione di sforzo normale nullo permette di ottenere soluzioni significative poiché approssima il caso in cui il condotto considerato termina nel vuoto, Figura 2.1.

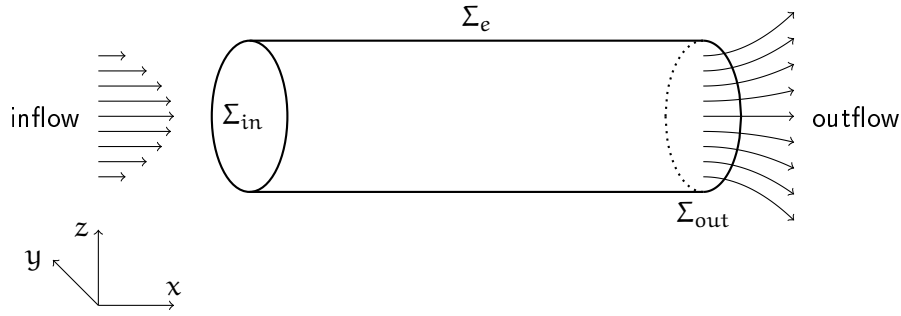


Figura 2.1: Condizioni naturali superficie di outflow,  $(2\mu\mathbf{S}(\mathbf{u}) - p\mathbf{I}) \cdot \mathbf{n} = 0$ .

## 2.1 Formulazione debole

La formulazione debole del problema (2.1) per  $d=3$  si ottiene moltiplicando le equazioni per una funzione test, rispettivamente  $\mathbf{v} \in \mathbf{V}$  e  $q \in \mathbf{Q}$ , e integrando poi sul dominio  $\Omega$ . Lo spazio scelto per la funzione  $\mathbf{v}$  è:

$$\mathbf{V} = [\mathbf{H}_{\Gamma_D}^1(\Omega)]^3 = \{\mathbf{v} \in [\mathbf{H}_{\Gamma_D}^1]^3 : \mathbf{v}|_{\Gamma_D} = 0\}.$$

La scelta dello spazio  $\mathbf{Q}$  verrà trattato successivamente, poiché sarà determinante al fine di garantire l'unicità della soluzione debole del problema.

La formulazione debole del problema (2.1) è:

trovare  $(\mathbf{u}, p) \in \mathbf{L}^2(\mathbb{R}^+; \mathbf{V}) \cap \mathbf{C}^0(\mathbb{R}^+; [\mathbf{L}^2(\Omega)]^3) \times \mathbf{L}^2(\mathbb{R}^+; \mathbf{Q})$ :

$$\begin{aligned} \int_{\Omega} \rho \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} \, d\mathbf{v} + \int_{\Omega} \rho (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} \, d\mathbf{v} - \int_{\Omega} \nabla \cdot (2\mu\mathbf{S}(\mathbf{u}) - p\mathbf{I}) \cdot \mathbf{v} \, d\mathbf{x} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}, \\ \int_{\Omega} \nabla \cdot \mathbf{u} \, q \, d\mathbf{x} &= 0 \quad \forall (\mathbf{v}, q) \in \mathbf{V} \times \mathbf{Q}. \end{aligned} \tag{2.3}$$

Applicando la formula di Green

$$\begin{aligned} \int_{\Omega} \nabla \cdot (2\mu\mathbf{S}(\mathbf{u}) - p\mathbf{I}) \cdot \mathbf{v} \, d\mathbf{x} &= - \int_{\Omega} 2\mu\mathbf{S}(\mathbf{u}) : \nabla \mathbf{v} \, d\mathbf{x} + \int_{\Omega} p \nabla \cdot \mathbf{v} \, d\mathbf{x} \\ &\quad + \int_{\partial\Omega} (2\mu\mathbf{S}(\mathbf{u}) - p\mathbf{I}) \cdot \mathbf{n} \cdot \mathbf{v} \, d\sigma, \end{aligned}$$

dove  $\mathbf{n}$  è la normale esterna al bordo.

La scelta dello spazio  $\mathbf{V}$  permette di annullare il termine di bordo sulla frontiera  $\Gamma_D$ .

Resta da annullare il termine:

$$\int_{\Gamma_N} (2\mu\mathbf{S}(\mathbf{u}) - p\mathbf{I}) \cdot \mathbf{n} \cdot \mathbf{v} \, d\sigma.$$



Per eliminare il contributo sul bordo  $\Gamma_N$  vengono imposte le condizioni naturali di sforzo nullo (2.2) e la (2.3) diventa:

$$\int_{\Omega} \rho \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} \, dx + \int_{\Omega} \rho (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{v} \, dx + \int_{\Omega} 2\mu \mathbf{S}(\mathbf{u}) : \nabla \mathbf{v} \, dx - \int_{\Omega} p \nabla \cdot \mathbf{v} \, dx = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx, \quad (2.4)$$

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, q \, dx = 0 \quad \forall (\mathbf{v}, q) \in V \times Q.$$

Per semplificare la notazione si introducono le forme bilineari:

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} 2\mu \mathbf{S}(\mathbf{u}) : \nabla \mathbf{v} \, dx,$$

$$b(\mathbf{u}, q) = - \int_{\Omega} \nabla \cdot \mathbf{u} \, q \, dx,$$

$$c(\mathbf{u}, \mathbf{u}, \mathbf{v}) = \int_{\Omega} \rho (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} \, dx,$$

e il funzionale lineare  $F(\cdot)$  associato al problema:

$$F(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx.$$

È ora possibile scrivere il sistema (2.4) come:

$$\text{trovare } (\mathbf{u}, p) \in L^2(\mathbb{R}^+; V) \cap C^0(\mathbb{R}^+; [L^2(\Omega)]^3) \times L^2(\mathbb{R}^+; Q):$$

$$\int_{\Omega} \rho \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} \, dx + a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = F(\mathbf{v}), \quad (2.5)$$

$$b(\mathbf{u}, q) = 0 \quad \forall (\mathbf{v}, q) \in V \times Q.$$

L'esistenza della soluzione di (2.5) è garantita dall'analisi svolta in [16].

Si discute ora la scelta per lo spazio test  $Q$  che è direttamente collegata all'unicità della soluzione.

Nel caso di condizioni al bordo full-Dirichlet,  $\partial\Omega = \Gamma_D$ , la pressione è definita a meno di una costante. Nella formulazione forte del problema (2.1) infatti la pressione compare sempre sotto il segno di derivata. Se quindi  $p$  è soluzione del problema, anche  $p + k$ , con  $k = \text{costante}$ , è soluzione del problema.

Le scelte possibili per evitare questo inconveniente sono quelle di fissare il valore

della pressione in un punto del dominio o richiedere che la pressione sia a media nulla. Richiedere che la pressione abbia un valore fissato in un determinato punto del dominio è però inconsistente con la scelta dello spazio  $L^2(\Omega)$ . Lo spazio delle funzioni a quadrato integrabile infatti non contiene necessariamente funzioni continue. Per questo motivo la pressione potrebbe quindi non ammettere valori puntuali. Dunque si sceglie di utilizzare lo spazio  $Q$  delle funzioni a media nulla per la pressione

$$Q = L_0^2(\Omega) = \left\{ q \in L^2(\Omega) : \int_{\Omega} p \, dx = 0 \right\}.$$

Nel caso di condizioni full-Neumann,  $\partial\Omega = \Gamma_N$ , scegliendo  $Q = L^2(\Omega)$ , è possibile garantire l'unicità della soluzione del problema debole.

## 2.2 Formulazione di Galerkin

Il passo successivo è la discretizzazione del problema. Si introduce una triangolazione  $\mathcal{T}_h$  di tetraedri  $K_j$  del dominio  $\Omega$  regolare e conforme e si considerano i sottospazi  $V_h \subset V$  e  $Q_h \subset Q$ , a dimensione finita, come nuova ambientazione per il problema. I due sottospazi sono dipendenti dal passo di discretizzazione  $h$ , non necessariamente costante, utilizzato nella costruzione di  $\mathcal{T}_h$ .

La formulazione di Galerkin del problema è dunque

per ogni  $t > 0$ , trovare  $(\mathbf{u}_h, p_h) \in V_h \times Q_h$ :

$$\int_{\Omega} \rho \frac{\partial \mathbf{u}_h}{\partial t} \cdot \mathbf{v}_h \, dx + a(\mathbf{u}_h, \mathbf{v}_h) + c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = F(\mathbf{v}_h), \quad (2.6)$$

$$b(\mathbf{u}_h, q_h) = 0 \quad \forall (\mathbf{v}_h, q_h) \in V_h \times Q_h.$$

L'esistenza e unicità della soluzione di (2.6) sono garantite dall'analisi svolta in [16] e dalla scelta di coppie  $(V_h, Q_h)$  compatibili, come verrà mostrato nel paragrafo successivo.

Si considerino ora le funzioni di base  $\{\boldsymbol{\varphi}_j \in V_h\}$  e  $\{\psi_k \in Q_h\}$ .

È possibile esprimere la soluzione del problema (2.6) come combinazione lineare di queste funzioni

$$\begin{aligned} \mathbf{u}_h(\mathbf{x}) &= \sum_{j=1}^{N_h} u_j \boldsymbol{\varphi}_j(\mathbf{x}) & N_h &= \dim V_h, \\ p_h(\mathbf{x}) &= \sum_{k=1}^{M_h} p_k \psi_k(\mathbf{x}) & M_h &= \dim Q_h. \end{aligned} \quad (2.7)$$

Poiché  $\mathbf{v}_h \in V_h$  e  $q_h \in Q_h$  è possibile scegliere  $\mathbf{v}_h = \boldsymbol{\varphi}_j$  e  $q_h = \psi_k$ .

Il problema algebrico corrispondente a (2.6) è:

$$\begin{cases} M \frac{\partial \mathbf{u}(t)}{\partial t} + \mathbf{A} \mathbf{u}(t) + \mathbf{C}(\mathbf{u}(t)) \mathbf{u}(t) + \mathbf{B}^T \mathbf{p}(t) = \mathbf{f}(t), \\ \mathbf{B} \mathbf{u}(t) = 0, \end{cases} \quad (2.8)$$

dove gli elementi delle matrici  $M \in \mathbb{R}^{N_h \times N_h}$ ,  $\mathbf{A} \in \mathbb{R}^{N_h \times N_h}$ ,  $\mathbf{B} \in \mathbb{R}^{M_h \times N_h}$  e  $\mathbf{C} \in \mathbb{R}^{N_h \times N_h}$  sono:

$$\begin{aligned} M &= [m_{ij}] = (\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) = \int_{\Omega} \boldsymbol{\varphi}_j \boldsymbol{\varphi}_i \, dx, \\ \mathbf{A} &= [a_{ij}] = [\mathbf{a}(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i)], \\ \mathbf{B} &= [b_{ij}] = [\mathbf{b}(\boldsymbol{\varphi}_j, \psi_i)], \\ \mathbf{C}(\boldsymbol{\beta}(t)) &= [c_{ii}(t)] = [c(\boldsymbol{\beta}(t), \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_i)]. \end{aligned}$$

La matrice  $M$  prende il nome di matrice di massa.

La risoluzione del problema di Navier-Stokes è quindi ricondotta al determinare, al variare del tempo, i vettori  $\mathbf{u}(t)$  e  $\mathbf{p}(t)$  soluzioni del sistema algebrico (2.8).

## 2.3 Il metodo di Galerkin a elementi finiti

Si introduce la famiglia di spazi

$$X_h^r = \{\mathbf{v}_h \in C^0(\overline{\Omega}) : \mathbf{v}_h|_{K_j} \in \mathbb{P}_r, \forall K_j \in \mathcal{T}_h\},$$

dove  $\mathbb{P}_r$  è lo spazio dei polinomi di grado uguale o inferiore a  $r$ .

Il metodo di Galerkin a elementi finiti è caratterizzato da una scelta delle basi  $\{\boldsymbol{\varphi}_i\}$  e  $\{\psi_k\}$  appartenenti alla famiglia  $X_h^r$ .

Gli spazi  $V_h$  e  $Q_h$  saranno quindi spazi polinomiali e le basi utilizzate polinomi definiti sui singoli elementi della triangolazione  $\mathcal{T}_h$ .

Un'altra caratteristica del metodo agli elementi finiti è l'utilizzo di un supporto per la funzione di base generica  $\psi_k$  ad intersezione non vuota con un numero limitato di altre funzioni di base. In questo modo molti elementi delle matrici che caratterizzano il problema algebrico saranno nulli. Le stesse proprietà vengono garantite per le basi vettoriali  $\boldsymbol{\varphi}_j$ .

Le basi considerate sono *lagrangiane*, per cui coefficienti  $\mathbf{u}_j$  e  $\mathbf{p}_k$  in (2.7) saranno i valori assunti dalle soluzioni in punti, chiamati *nod*i, che apparterranno ad un sovrainsieme dei vertici dei tetraedri  $K_j$  della triangolazione  $\mathcal{T}_h$ . La Figura 2.2 mostra un esempio di basi nel caso monodimensionale.

Classici spazi agli elementi finiti sono gli spazi delle funzioni costanti a tratti,



**Figura 2.2:** Funzioni di base per il nodo  $x_i$ . Elementi di  $X_h^1$  a sinistra, di  $X_h^2$  a destra

lineari e quadratiche che indicheremo rispettivamente con  $\mathcal{P}_0$ ,  $\mathcal{P}_1$  e  $\mathcal{P}_2$ . Affinché la soluzione del problema di Galerkin sia unica è necessario che sia soddisfatta la condizione *LBB* [3]. Nel caso del problema ad elementi finiti, questa condizione è garantita dalla particolare scelta di *coppie compatibili* degli spazi  $V_h$  e  $Q_h$ . L'utilizzo di coppie non compatibili introduce nel problema il fenomeno dei *modi spuri* di pressione, che rendono quest'ultima non unica.

Le coppie compatibili per il problema su griglie tetraedriche sono:

$$(V_h, Q_h) = (\mathcal{P}_n, \mathcal{P}_{n-1}) \quad n > 2.$$

Un'ulteriore coppia compatibile è quella

$$(V_h, Q_h) = (\mathcal{P}_{1b}, \mathcal{P}_1),$$

dove il simbolo  $\mathcal{P}_{1b}$  rappresenta i cosiddetti *elementi finiti lineari bolla*.

Gli elementi  $\mathcal{P}_{1b}$  possono essere interpretati come elementi finiti lineari arricchiti, in cui lo spazio  $V_h$  viene sostituito dallo spazio  $V_h^b$

$$V_h^b = V_h + B$$

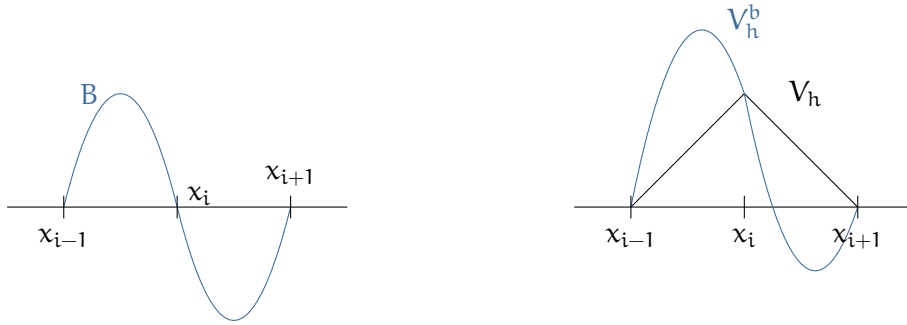
dove  $B$  è lo spazio finito dimensionale delle *funzioni a bolla*

$$B = \{v_b \in V : v_b|_K = c_K b_K, b_K|_{\partial K} = 0, c_K \in \mathbb{R}\}.$$

Su ogni elemento viene quindi aggiunto un termine correttivo  $b_k$  che stabilizza il problema. La Figura 2.3 mostra un esempio monodimensionale di una possibile scelta per lo spazio  $B$ .

## 2.4 Discretizzazione temporale

Dopo aver effettuato la discretizzazione spaziale del problema di Navier-Stokes, tramite il metodo di Galerkin, fino ad ottenere il problema algebrico (2.8), se ne



**Figura 2.3:** Funzioni di base di tipo bolla a sinistra, elementi di  $\mathcal{P}_{1b}$  a destra.

mostra ora la discretizzazione temporale.

L'approssimazione utilizzata per il termine evolutivo

$$M \frac{\partial \mathbf{u}(t)}{\partial t},$$

da come risultato il sistema

$$\begin{aligned} \rho \frac{\alpha_0}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} \, dx + \rho \int_{\Omega} (\mathbf{u}^* \cdot \nabla) \mathbf{u}^{n+1} \cdot \mathbf{v} \, dx \\ + 2\mu \int_{\Omega} \mathbf{S}(\mathbf{u}^{n+1}) : \nabla \mathbf{v} \, dx - \int_{\Omega} p^{n+1} \nabla \cdot \mathbf{v} \, dx = \\ \rho \int_{\Omega} \mathbf{f}_V^{n+1} \cdot \mathbf{v} \, dx + \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{v} \, dx, \end{aligned} \quad (2.9)$$

$$\int_{\Omega} \nabla \cdot \mathbf{u}^{n+1} q \, dx = 0 \quad \forall (\mathbf{v}_h, q_h) \in V_h \times Q_h,$$

dove  $\Delta t$  è il passo temporale,  $\alpha_0$  e  $\mathbf{f}_V$  dipendono dallo schema scelto per l'avanzamento temporale.

Dalla scelta di  $\mathbf{u}^*$  dipendono alcune proprietà del problema. In questo lavoro utilizziamo una discretizzazione semi-implicita del termine convettivo, ovvero  $\mathbf{u}^* = \mathbf{u}^n$ . Di conseguenza il problema di Navier-Stokes (2.1) si riconduce al problema di Oseen. Si definisce inoltre il vettore di stato  $\mathbf{X}^{n+1}$  che contiene le soluzioni relativi agli istanti temporali precedenti.

Si presentano di seguito due famiglie di schemi:

**Backward Difference (BDF)** : il vettore  $\mathbf{X}^{n+1}$  corrisponde a

$$\mathbf{X}^{n+1} = (\mathbf{u}^{n+1}, \mathbf{u}^n, \dots, \mathbf{u}^{n+1-s}),$$

dove  $s$  è l'ordine del metodo.

Il termine  $\mathbf{f}_V^{n+1}$  in (2.9) è

$$\mathbf{f}_V^{n+1} = \sum_{i=1}^s \frac{\alpha_i}{\Delta t} \mathbf{u}^{n+1-s},$$

con i coefficienti  $\alpha_i$  determinati da

$$\frac{\partial \mathbf{Y}}{\partial t} \Big|_{t=t^{n+1}} = \frac{\alpha_0}{\Delta t} \mathbf{Y}(t^{n+1}) - \sum_{i=1}^s \frac{\alpha_i}{\Delta t} \mathbf{Y}(t^{n+1-s}) + O(\Delta t^s) \quad \forall \mathbf{Y}(t).$$

Nel caso di una discretizzazione semi-implicita il termine  $\mathbf{u}^*$  corrisponde all'estrapolazione  $\mathbf{w}^*$

$$\mathbf{w}^* = \sum_{i=0}^s \omega_i \mathbf{u}^{n-i} = \mathbf{u}^{n+1} + O(\Delta t^s),$$

dove  $\omega_i$  sono coefficienti appropriati. È ora possibile introdurre una approssimazione consistente per  $\frac{\partial \mathbf{u}}{\partial t}$  con lo schema

$$\lambda^{n+1} = \frac{\alpha_0}{\Delta t} \mathbf{u}^{n+1} - \mathbf{f}_V^{n+1}.$$

**$\theta$ -metodo** : il vettore  $\mathbf{X}^{n+1}$  corrisponde a

$$\mathbf{X}^{n+1} = (\mathbf{u}^{n+1}, \lambda^{n+1}, \mathbf{u}^n, \lambda^n),$$

dove  $\lambda^{n+1}$ , approssimazione di  $\frac{\partial \mathbf{u}}{\partial t}$ , è valutata come segue

$$\lambda^{n+1} = \frac{1}{\theta \Delta t} (\mathbf{u}^{n+1} - \mathbf{u}^n) + (1 - \frac{1}{\theta}) \lambda^n, \quad \theta \in (0, 1].$$

Le quantità presenti in (2.9) sono

$$\alpha_0 = \frac{1}{\theta}, \quad \mathbf{f}_V^{n+1} = \frac{1}{\theta \Delta t} \mathbf{u}^n + (\frac{1}{\theta} - 1) \lambda^n.$$

Utilizzando una discretizzazione semi-implicita del termine convettivo di Navier-Stokes,  $\mathbf{u}^* = \mathbf{w}^*$ , si utilizza l'estrapolazione

$$\mathbf{w}^* = \mathbf{u}^n + \Delta t \lambda^n.$$

La formulazione algebrica del sistema (2.9) è quindi

$$\begin{cases} \frac{\alpha_0}{\Delta t} \mathbf{M} \mathbf{u}^{n+1} + \mathbf{A} \mathbf{u}^{n+1} + \mathbf{C}(\mathbf{u}^*) \mathbf{u}^{n+1} + \mathbf{B}^T \mathbf{p}^{n+1} = \mathbf{f}^{n+1} + \mathbf{M} \mathbf{f}_V^{n+1}, \\ \mathbf{B} \mathbf{u}^{n+1} = 0. \end{cases} \quad (2.10)$$

## 2.5 Il termine di sottogriglia

Ci si occupa ora del termine

$$\rho \frac{\partial \tau_{ij}}{\partial x_j},$$

che si ottiene dal filtraggio delle equazioni di Navier-Stokes (2.1) utilizzando la procedura descritta nel Capitolo 1.

Dopo aver moltiplicato per una funzione test, integrato sul dominio  $\Omega$  e applicato la formula di Green al problema (2.1) filtrato si ottiene

$$\begin{aligned} \int_{\Omega} \rho \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} \, dx + \int_{\Omega} \rho (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{v} \, dx + \int_{\Omega} 2\mu \mathbf{S}(\mathbf{u}) : \nabla \mathbf{v} \, dx \\ - \int_{\Omega} p \nabla \cdot \mathbf{v} \, dx - \int_{\Omega} \rho \boldsymbol{\tau}(\mathbf{u}) : \nabla \mathbf{v} \, dx = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx, \end{aligned} \quad (2.11)$$

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, q \, dx = 0 \quad \forall (\mathbf{v}, q) \in V \times Q.$$

Il problema algebrico corrispondente è

$$\begin{cases} M \frac{\partial \mathbf{u}(t)}{\partial t} + \mathbf{A} \mathbf{u}(t) + \mathbf{C}(\mathbf{u}(t)) \mathbf{u}(t) + \mathbf{B}^T \mathbf{p}(t) - \mathbf{G}(\mathbf{u}(t)) \mathbf{u}(t) = \mathbf{f}(t), \\ \mathbf{B} \mathbf{u}(t) = 0, \end{cases} \quad (2.12)$$

dove la matrice  $\mathbf{G}$  corrisponde a

$$\mathbf{G} = [\gamma_{ij}] = \int_{\Omega} \boldsymbol{\tau}(\varphi_j) : \nabla \varphi_i \, dx.$$

La dipendenza da  $\mathbf{u}$  della matrice  $\mathbf{G}$  è dovuta all'applicazione del modello di Smagorinsky al termine di sottogriglia.

Il termine di sottogriglia viene calcolato in forma semi-implicita, il sistema (2.10) diventa quindi

$$\begin{cases} \frac{\alpha_0}{\Delta t} M \mathbf{u}^{n+1} + \mathbf{A} \mathbf{u}^{n+1} + \mathbf{C}(\mathbf{u}^*) \mathbf{u}^{n+1} + \mathbf{B}^T \mathbf{p}^{n+1} - \mathbf{G}(\mathbf{u}^n) \mathbf{u}^{**} = \mathbf{f}^{n+1} + M \mathbf{f}_V^{n+1}, \\ \mathbf{B} \mathbf{u}^{n+1} = 0. \end{cases} \quad (2.13)$$

La forma del termine  $\boldsymbol{\tau}$  e di conseguenza della matrice  $\mathbf{G}$  dipenderà fortemente dal tipo di modello utilizzato.

Per i modelli utilizzati in questo lavoro,  $\mathbf{G}$  ha le seguenti forme:

- Modello di Smagorinsky:

$$\mathbf{G}(\mathbf{u}^n) = [\gamma_{ij}] = \int_{\Omega} -2C_S^2 \Delta^2 |\mathbf{S}(\mathbf{u}^n)| \mathbf{S}(\varphi_j) : \nabla \varphi_i \, dx. \quad (2.14)$$

- Modello di Smagorinsky con approssimazione di Van Driest:

$$G(\mathbf{u}^n) = [\gamma_{ij}] = \int_{\Omega} -2 \left[ C_S \Delta (1 - e^{-y^+/\Lambda}) \right]^2 |\mathbf{S}(\mathbf{u}^n)| \mathbf{S}(\varphi_j) : \nabla \varphi_i \, dx. \quad (2.15)$$

- Modello di Germano:

$$G(\mathbf{u}^n) = [\gamma_{ij}] = \int_{\Omega} -2 \left[ C_S(\mathbf{u}^n) \Delta (1 - e^{-y^+/\Lambda}) \right]^2 |\mathbf{S}(\mathbf{u}^n)| \mathbf{S}(\varphi_j) : \nabla \varphi_i \, dx.$$

La trattazione a elementi finiti del termine di Germano si può trovare in [11].



## Implementazione in LifeV

Lo scopo di questa tesi è l'introduzione nella libreria degli strumenti per la trattazione della turbolenza tramite la tecnica LES.

Il lavoro è stato svolto a partire da un caso test presente in LifeV a cui sono stati aggiunti metodi e classi specifici.

Di seguito verranno descritti gli elementi del codice utilizzati e le novità introdotte.

### 3.1 Griglia

Utilizzando i software specifici *gms* [6] e *Freefem++* [7] la griglia di calcolo viene costruita con la geometria e il numero di nodi desiderati. Il risultato di questa prima operazione viene modificato grazie a un programma creato appositamente, scritto in linguaggio C che permette di modificare le dimensioni della griglia, di raffinarla e deraffinarla per renderla adatta al problema considerato.

Una volta avviata la simulazione, la griglia viene letta e partizionata per essere distribuita sui processori e permettere così il calcolo in parallelo. Lo strumento utilizzato per questa procedura è ParMETIS, libreria MPI specializzata nel partizionamento di grafi, griglie e matrici sparse.

Il limite principale di questo procedimento è relativo alle dimensioni della griglia. Griglie molto grandi distribuite su un numero piccolo di processori possono dare luogo a simulazioni poco efficienti. Questa distribuzione porta infatti a calcoli troppo onerosi per il singolo processore. Un altro possibile motivo di inefficienza può essere l'utilizzare un numero di processori troppo elevato per una griglia di dimensioni ridotte. Il tempo di comunicazione tra i processori sarà maggiore del tempo effettivo di calcolo e la simulazione ne risentirà in termini di prestazioni.

La classe responsabile del partizionamento è:

```
MeshPartitioner< RegionMesh3D<LinearTetra> >
```

La classe riceve una griglia di tipo `RegionMesh3D<LinearTetra>` e chiama `ParMETIS` per partizionarla e distribuirla sui processori utilizzati per la simulazione.

## 3.2 Matrici e Vettori

La gestione di matrici e vettori in `LifeV` avviene tramite il pacchetto `Epetra` [8], in cui le matrici sono `Epetra_FE_CrsMatrix` e i vettori `Epetra_FE_Vector`.

Le componenti dei vettori e delle matrici, vengono distribuite in funzione del partizionamento della griglia. Un ciclo sugli elementi di griglia assegna le componenti ai singoli processori e viene creata una mappa *ripetuta* `MapEpetra`. Successivamente viene creata una mappa *unica*. La prima permette l'accesso alle informazioni condivise tra più processori, nodi, spigoli e facce all'interfaccia, mentre la seconda è utilizzata nella risoluzione delle operazioni tra vettori e matrici.

L'assemblaggio delle matrici che compongono il sistema lineare da risolvere viene fatto localmente sui singoli elementi. Quando la costruzione locale è terminata viene chiamato il metodo `GlobalAssemble` che costruisce la matrice globale per il sistema. La matrice costruita non può più essere modificata.

L'accesso alle classi di `Epetra` in `LifeV` avviene tramite classi interfaccia come la classe `MatrixEpetra` per le matrici, la classe `VectorEpetra` per i vettori e la classe `MapEpetra` per le mappe.

Per permettere la comunicazione tra le classi interfaccia e le classi di `Epetra` `LifeV` utilizza la classe virtuale `Epetra_Comm`.

Gli `Epetra_Comm` sono utilizzati per la costruzione di tutte le mappe necessarie alla creazione di oggetti di tipo `Epetra`, matrici e vettori. Il codice permette due possibili implementazioni, seriale e parallela.

Definito il *comunicatore* è possibile creare la mappa, `MapEpetra`, responsabile della suddivisione delle informazioni tra i vari processori per permettere il calcolo in parallelo.

La creazione di una mappa permette di costruire le classi interfaccia `VectorEpetra` e `MatrixEpetra`.

Le classi per la gestione di matrici e vettori utilizzano costruttori forniti da `Trilinos` che permettono anche di importare gli elementi del vettore da un file di dati. Sono inoltre disponibili due tipologie di matrici, una *seriale* di piccole dimensioni utilizzata

localmente, e una *distribuita* su più processori.

Le classi interfaccia permettono tutte le più comuni operazioni algebriche tra vettori e matrici tramite gli operatori  $+$ ,  $+=$ ,  $-$ ,  $-=$ , e la moltiplicazione per uno scalare con gli operatori  $*$ ,  $*=$ . Sono possibili anche operazioni più avanzate quali il calcolo della norma 2 o della norma infinito, il calcolo della media o del valore massimo di un vettore. Gli operatori  $()$ ,  $[]$  permettono l'accesso ai singoli elementi del vettore o della matrice.

Le classi `VectorEpetra.hpp` e `MatrixEpetra.hpp` sono inoltre fornite di metodi per l'accesso e la restituzione dei dati, di `Import` e `Export` e di metodi che lavorando con le mappe permettono un collegamento tra ambiente locale, il singolo elemento, e ambiente globale, l'intera griglia.

Le classi finora presentate si occupano della costruzione di matrici e vettori sulla griglia partizionata tra i vari processori.

Della gestione locale, sul singolo elemento, di vettori e matrici si occupano invece le classi `MatrixElemental` e `VectorElemental`. Come verrà mostrato in seguito le matrici saranno infatti costruite localmente e solo in seguito assemblate.

### 3.3 FESpace

La classe `FESpace` rappresenta uno spazio a elementi finiti. La classe contiene l'elemento di riferimento e la mappa che permette il passaggio all'elemento specifico e viceversa.

Lo spazio a elementi finiti viene costruito passando al costruttore informazioni relative a:

- griglia,
- dimensioni del problema,
- elementi finiti di riferimento,
- regole di quadratura per l'integrazione numerica,
- condizioni al bordo,
- schema di avanzamento temporale.

Superata la sezione relativa ai costruttori, percorrendo il file della classe `FESpace`, si incontrano metodi che implementano alcune operazioni importanti per l'analisi dei dati e per la costruzione algebrica del sistema da risolvere.

Sono presenti metodi per l'interpolazione nodale di funzioni, come `interpolate` e `interpolateBC`.

Viene mostrata di seguito la firma del metodo `l2ScalarProduct`:

```
// this computes vec = \int fct phi_i
template<typename vector_type>
void l2ScalarProduct ( const function_Type& fct ,
                       vector_type& vec ,
                       const Real t );
```

Il metodo, che verrà utilizzato per calcolare la forzante del problema di Oseen, esegue il prodotto scalare nello spazio  $L^2$  tra la funzione `fct` e la base dello spazio a elementi finiti al tempo `t`. Il risultato dell'operazione viene salvato nel vettore `vec`.

I metodi di analisi dei dati permettono di calcolare norme, di funzioni e vettori nodali, ed errori negli spazi  $L^2$ ,  $L_0^2$  e  $H^1$ .

### 3.4 IFPACK e Aztec00

Per la soluzione dei sistemi algebrici viene utilizzato il pacchetto IFPACK di Trilinos [8]. Questo pacchetto fornisce una serie di preconditionatori algebrici per la risoluzione iterativa di sistemi. Il codice LifeV permette di scegliere a priori i parametri da utilizzare nella risoluzione del sistema quali il numero di sovrapposizioni, `overlap`, e il tipo di solutori locali (LU, ILU).

Scelto il preconditionatore per la matrice del sistema il pacchetto Aztec00 [8] fornisce i metodi iterativi (GMRES, BiCGStab, CG) per la risoluzione del problema.

### 3.5 Il problema di Oseen turbolento

Il problema affrontato in questa tesi è il problema di Oseen con un termine aggiuntivo associato alla turbolenza, come mostrato nelle sezioni precedenti.

La formulazione del problema di Oseen è trattata in LifeV per un dominio cilindrico. Attraverso i metodi della classe `OseenSolver` il problema algebrico viene costruito e risolto. È stata inoltre aggiunta la trattazione del problema della corrente nel canale piano.

La classe `OseenSolver` chiama i metodi presenti in `AssemblyElemental` per la costruzione locale dei termini da assemblare per formare le matrici globali del sistema.

In questo lavoro sono state introdotte due nuove funzioni per la costruzione locale

dei termini di turbolenza in `AssemblyElemental`.

Alcuni metodi di `OseenSolver` sono stati modificati per permettere, a discrezione dell'utente, di utilizzare o meno i modelli di turbolenza introdotti.

Verranno mostrate nell'ordine:

- le funzioni utilizzate nella costruzione dei termini locali del sistema, facendo particolare attenzione alle nuove funzioni introdotte,
- i metodi di `OseenSolver` che assemblano e risolvono il sistema globale,
- il file `cylinder.cpp` che definisce un problema specifico mediante l'assegnazione delle condizioni al bordo e lo risolve chiamando i metodi introdotti in precedenza,
- il data-file che permette all'utente di scegliere i parametri della simulazione.

### 3.5.1 `AssemblyElemental`

I file `AssemblyElemental.hpp` e `AssemblyElemental.cpp` contengono le procedure per la costruzione locale dei diversi operatori differenziali.

Si mostra ora come attraverso i metodi contenuti in questi file è possibile costruire le strutture locali del problema (2.10).

La prima matrice costruita è la matrice di dissipazione

$$A = [a_{ij}] = [a(\varphi_j, \varphi_i)] = 2\mu \int_{\Omega} S(\varphi_j) : \nabla \varphi_i \, dx. \quad (3.1)$$

A questa matrice corrisponde il metodo

```
void stiff( Real coef, MatrixElemental& elmat,
           const CurrentFE& fe,
           int iblock, int jblock, int nb );
```

nel caso in cui  $S(\mathbf{u}) = \nabla \mathbf{u}$ , mentre

```
void stiff_strain( Real coef, MatrixElemental& elmat,
                  const CurrentFE& fe );
```

se  $S(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla^T \mathbf{u})$ .

L'attributo `coef` è il coefficiente per cui è moltiplicato l'integrale in (3.1), `elmat` è la matrice locale che verrà riempita con i valori corrispondenti al termine di diffusione, mentre `fe` è l'elemento finito locale considerato. `iblock` e `jblock` sono attributi utilizzati per la localizzazione del singolo blocco della matrice, nel caso in cui quest'ultima sia una sotto matrice di una matrice elementare di dimensioni maggiori. L'attributo

nb dà le dimensioni della matrice.

La matrice di massa:

$$M = [m_{ij}] = (\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) = \int_{\Omega} \boldsymbol{\varphi}_j \boldsymbol{\varphi}_i \, dx,$$

viene costruita attraverso il metodo

```
void mass( Real coef, MatrixElemental& elmat,
           const CurrentFE& fe,
           int iblock, int jblock, UInt nb );
```

La matrice che si occupa del gradiente di pressione:

$$B = [b_{ij}] = [b(\boldsymbol{\varphi}_j, \psi_i)] = - \int_{\Omega} \nabla \cdot \boldsymbol{\varphi}_j \psi_i \, dx,$$

viene costruita in un ciclo for nella classe OseenSolver.hpp che verrà mostrato nella prossima sezione. Il metodo utilizzato all'interno di questo ciclo è grad.

```
void grad( const int icoor, Real coef,
           MatrixElemental& elmat,
           const CurrentFE& fe_u,
           const CurrentFE& fe_p,
           int iblock, int jblock )
```

La singola chiamata di questo metodo infatti esegue le seguenti operazioni

$$- \int_{\Omega} \psi \frac{\partial \boldsymbol{\varphi}_{i\text{coor}}}{\partial x_{i\text{coor}}} \, dx.$$

La chiamata ricorsiva permetterà quindi la costruzione della matrice B.

Per poter costruire il sistema che risolve il problema di Oseen resta da costruire il termine convettivo che corrisponde in forma debole alla matrice:

$$C(\boldsymbol{\beta}(t)) = [c_{li}(t)] = [c(\boldsymbol{\beta}(t), \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_l)] = \int_{\Omega} (\boldsymbol{\beta}(t) \cdot \nabla \boldsymbol{\varphi}_i) \cdot \boldsymbol{\varphi}_l \, dx.$$

La funzione corrispondente è advection di cui di seguito viene mostrata la firma

```
void advection( Real coef,
               VectorElemental& vel,
               MatrixElemental& elmat,
               const CurrentFE& fe,
               int iblock, int jblock, int nb )
```

Gli attributi passati alle funzioni mostrate hanno il medesimo significato di quelli passati alla funzione stiff.

Gli ulteriori attributi che compaiono sono fe\_p che corrisponde all'elemento finito

locale per lo spazio di pressione e il vettore `vel` che contiene i valori nodali del vettore  $\beta$ , risultato dell'estrapolazione numerica nello schema di avanzamento temporale. Le funzioni mostrate di seguito rappresentano le novità introdotte nel codice. Queste funzioni sono direttamente responsabili del calcolo dei termini (2.14) e (2.15). Le firme delle due funzioni sono rispettivamente:

```
void subGridSmagorinsky( Real coef ,
                        MatrixElemental& elmat ,
                        VectorElemental& elvec_u ,
                        const CurrentFE& fe )
```

e

```
void subGridVanDriest( Real coef ,
                      MatrixElemental& elmat ,
                      VectorElemental& elvec_u ,
                      const CurrentFE& fe ,
                      const Real ReT )
```

Gli attributi `coef`, `elmat` e `fe` sono come nei casi precedenti, il coefficiente che moltiplica la matrice, la matrice locale e l'elemento finito per lo spazio della velocità. `ReT` rappresenta il numero di Reynolds del problema, mentre `elvec_u` è il vettore che contiene il valore della velocità nei nodi al passo temporale precedente. Verrà illustrato in particolare il codice della funzione `subGridVanDriest` che implementa alcune operazioni in più rispetto al metodo `subGridSmagorinsky`.

### La funzione `subGridVanDriest`

Come mostrato in precedenza la correzione di Van Driest è una procedura applicata al metodo di Smagorinsky per evitare la sovradiffusione che lo caratterizza nello strato limite. Nelle prime righe del codice viene calcolato il contributo locale del tensore

$$|S(\mathbf{u}^n)| = \sqrt{2S_{ij}S_{ij}}.$$

A partire dai valori nodali contenuti in `elvec_u`, il codice calcola il gradiente della velocità al passo precedente e lo salva nel vettore `grad`.

I valori così ottenuti vengono sommati con il rispettivo elemento della matrice traspota e elevati al quadrato.

L'ultima riga del codice esegue la radice quadrata e sistema i coefficienti, salvando il valore di  $|S|$  nei nodi di quadratura nel vettore `strainNorm`:

```
std::vector<Real> grad(nDim*nDim);
```

```

std::vector<Real> strainNorm(fe.nbQuadPt());

//shear norm computing \sqrt{2} e(u) : e(u) }
for ( int iq = 0; iq < fe.nbQuadPt(); iq++ )
{
    strainNorm[iq]=0;

    for ( int ic = 0; ic < (int)nDim; ic++ )
    {
        for ( int jc = 0; jc < (int)nDim; jc++ )
        {
            Real s(0);
            for ( int i = 0; i < fe.nbfEDof(); i++ )
            {
                s += elvec_u.vec()[i+ic*fe.nbfEDof()]
                    *fe.phiDer( i, jc, iq );
            }
            grad[ic*nDim+jc] = s;
        }
    }

    for ( UInt ic = 0; ic < nDim; ic++ )
    {
        for ( UInt jc = 0; jc < nDim; jc++ )
        {
            strainNorm[iq]+=pow(grad[ic*nDim+jc]+grad[jc*nDim+ic],2);
        }
    }
    strainNorm[iq] = sqrt(2 * shear_rate[iq])/2;
}

```

La parte successiva della funzione si occupa di calcolare il valore del parametro  $\Delta$  corrispondente all'ampiezza del filtro e del termine di correzione

$$(1 - e^{-y^+/\Delta}).$$

L'ampiezza del filtro può essere calcolata come la radice cubica del volume dell'elemento in cui si sta operando o come il suo diametro in norma 2. Il valore calcolato viene salvato in `DxDyDz`. Nella sezione dei risultati verranno confrontati risultati ottenuti con i differenti approcci.

Le coordinate `x`, `y` e `z` vengono in seguito inizializzate con il baricentro dell'elemento `fe`.

```
double s, dist;
```



```
// \Delta = volume cube root
// double DxDyDz=cbrt( fe.measure() );
// \Delta = fe diameter in 2 norm
double DxDyDz=fe.diameter2() ;
Real x,y,z,value;
fe.barycenter(x,y,z);
```

Successivamente la funzione calcola la distanza dal bordo  $y^+$ , verificando che non sia negativa, situazione che si può verificare poiché il dominio cilindrico è composto da spezzate, e costruisce il termine di correzione salvandolo nella variabile `dist`. Infine viene calcolato il valore del coefficiente di viscosità turbolenta a meno della costante di Smagorinsky, che verrà moltiplicata successivamente per la matrice. Il coefficiente ottenuto è contenuto nella variabile `tmp`

```
//cylinder
dist= 0.5 - sqrt( x*x + y*y );
if (dist < 0.) dist=0.;

dist*= (-ReT/(25));
dist=(1. - exp(dist));

value=DxDyDz * DxDyDz * dist * dist;

double tmp = coef * 0.5 * value;
```

Queste operazioni sono implementate anche per la corrente di Poiseuille, con la differenza che il valore della coordinata  $y^+$  è il minimo tra le distanze dalle pareti superiore e inferiore del dominio.

Una volta che i coefficienti sono valutati sull'elemento locale, si procede alla costruzione della parte differenziale della matrice e al suo assemblaggio tramite integrazione numerica. Questo procedimento è fatto tramite una serie di cicli `for` annidati di cui parte viene mostrata di seguito. Questi cicli sommano per ogni grado di libertà, nella variabile `s`, il prodotto del contributo delle derivate, del peso di quadratura e il valore nodale di  $|S|$ .

```
MatrixElemental::matrix_type mat_tmp(fe.nbFEDof(), fe.nbFEDof());

for ( UInt i = 0; i < fe.nbFEDof(); ++i )
{
    for ( UInt j = 0; j < fe.nbFEDof(); ++j )
    {
        s = 0;
        for ( UInt ig = 0; ig < fe.nbQuadPt(); ++ig )
```

```

    {
        for ( UInt icoor = 0; icoor < fe.nbCoor(); ++icoor )
        {
            s+=fe.phiDer(i, icoor, ig)*fe.phiDer(j, icoor, ig)
              * fe.weightDet( ig ) * strainNorm[ig];
        }
    }
    mat_tmp( i, j ) = tmp * s;
}
}
}

```

### 3.5.2 OseenSolver

Conclusa la costruzione locale per ogni elemento, le matrici globali possono essere assemblate. Queste operazioni sono effettuate nella classe `OseenSolver` utilizzando i metodi descritti nella sezione precedente.

Il costruttore della classe `OseenSolver` riceve in ingresso un puntatore `dataType` alla classe di tipo `oseenData` che è la responsabile della lettura dei dati. Gli altri parametri sono gli spazi a elementi finiti per velocità e pressione, il comunicatore e il valore del moltiplicatore di Lagrange, uguale a zero per il problema che si sta risolvendo.

```

OseenSolver( boost::shared_ptr<data_Type>    dataType ,
              FESpace<mesh_Type, MapEpetra>& velocityFESpace ,
              FESpace<mesh_Type, MapEpetra>& pressureFESpace ,
              boost::shared_ptr<Epetra_Comm>& communicator ,
              const Int                       lagrangeMultiplier = 0 );

```

Gli attributi della `OseenSolver` classe sono una serie di booleani per attivare o meno alcune funzioni o modificare la costruzione di alcune matrici e le matrici stesse. Per la loro costruzione sono utilizzate due tipologie di attributi. La classe fornisce un puntatore a matrice, `matrixPtr_Type`, che punterà la matrice globale del rispettivo termine. Una matrice di tipo `MatrixElemental` si occuperà invece di contenere i termini locali prima dell'assemblaggio da locale a globale.

```

//! mass matrix
matrixPtr_Type          M_matrixMass;
//! Stokes matrix: nu*stiff
matrixPtr_Type          M_matrixStokes;
//!Subgrid term tau_ij
matrixPtr_Type          M_matrixSubgrid;
//! Mass elemental matrix
MatrixElemental         M_elementMatrixMass;

```



```
const vector_Type& sourceVector);
```

La prima versione è il metodo standard che non introduce alcun termine di modellazione della turbolenza. Il parametro alpha è il coefficiente per l'avanzamento temporale, betaVector contiene i valori della velocità estrapolata e sourceVector i valori nodali del termine forzante.

La seconda versione del metodo ha dei parametri aggiuntivi che sono: il vettore uPrev che contiene i valori nodali della velocità al passo temporale precedente  $\mathbf{u}^n$  e una stringa che contiene le informazioni sul modello di turbolenza da implementare.

```
virtual void updateSystem( const Real      alpha ,
                          const vector_Type& betaVector ,
                          const vector_Type& uPrev ,
                          const vector_Type& sourceVector ,
                          const std::string model="none");
```

Entrambi i metodi richiamano la terza versione di updateSystem che è il responsabile della costruzione delle matrici C e G.

```
virtual void updateSystem( const Real      alpha ,
                          const vector_Type& betaVector ,
                          const vector_Type& uPrev ,
                          const vector_Type& sourceVector ,
                          matrixPtr_Type  matrix ,
                          matrixPtr_Type  matrixSubgrid ,
                          vectorPtr_Type  un ,
                          const std::string model);
```

Le due matrici passate come parametri sono quelle che conterranno i termini creati. Il vettore un contiene i valori nodali della velocità con una mappa di tipo Unique.

Il metodo updateSystem, costruite le matrici, si occupa del calcolo del preconditionatore richiamando un metodo del solutore lineare M\_linearSolver.

Il metodo iterate applica le condizioni al contorno alla matrice del sistema e con la chiamata al metodo solveSystem( ... ) risolve il sistema lineare per mezzo di un metodo iterativo.

```
virtual void iterate( bcHandler_Type& bcHandler );
```

Il parametro bcHandler si occupa della gestione delle condizioni al contorno.

Il metodo solution() ritorna il puntatore alla soluzione corrente del sistema.

```
const vectorPtr_Type& solution() const
{
    return M_solution;
```

```
}

```

### 3.5.3 Cylinder

In questa sezione si mostra il codice del file `cylinder.cpp` utilizzato per la costruzione e risoluzione del problema algebrico (2.13) per un dominio cilindrico. Il codice relativo alla corrente nel canale piano contiene delle piccole differenze. Le più importanti verranno mostrate all'interno di questa sezione.

La prima parte del file si occupa delle condizioni al bordo. Ad ogni superficie della griglia è associata un'etichetta numerica. Queste righe di codice importano le etichette in `LifeV` per permettere l'assegnazione delle condizioni al bordo.

Per il canale cilindrico:

```
const int INLET      = 2;
const int WALL       = 1;
const int OUTLET     = 3;
const int RINGIN     = 20;
const int RINGOUT    = 30;
```

mentre per il canale piano:

```
const int INLET      = 4;
const int OUTLET     = 2;
const int WALLDX     = 1;
const int WALLINF    = 5;
const int WALLSX     = 3;
const int WALLSUP    = 6;
```

Nel file seguono la definizione di alcuni parametri che verranno letti direttamente dal file `data` e delle funzioni che descrivono le condizioni iniziali e le condizioni al contorno.

Di seguito viene mostrata la funzione utilizzata per imporre il profilo iniziale logaritmico turbolento nel cilindro, lungo la direzione  $z$ :

$$\begin{cases} \mathbf{u}_x = 0, \\ \mathbf{u}_y = 0, \\ \mathbf{u}_z = 2.5 \log \rho^+ + 5.5 & \text{se } \rho^+ > 11.6, \\ \mathbf{u}_z = \rho^+ & \text{se } \rho^+ \leq 11.6. \end{cases} \quad (3.2)$$

```
Real initCond( const Real& t,
               const Real& x,
               const Real& y,
```

```

        const Real& z,
        const ID& id ) const
{
    int rndInt, resto;
    Real rndReal, somma;
    Real value=0.;

    //evaluation of wall distance
    double dist = ( D/2 - std::sqrt(x*x + y*y) ) * Re;
    if (dist < 0.) dist = 0.;

    //evaluation of random perturbation
    rndInt=std::rand();
    Real=(Real)rndInt;
    resto=rndInt%2;
    somma=(rndReal/RAND_MAX);
    if(!resto) somma *= -1;

    //velocity value
    if(dist > 11.6) value=2.5 * std::log(dist) + 5.5;
    else value = dist;

    switch(id)
    {
        case 0: // x coordinate
            value*= (pert * somma);
            return value;
            break;
        case 1: // y coordinate
            value*= (pert * somma);
            return value;
            break;
        case 2: // z coordinate
            value*= (1 + pert*somma);
            return value;
            break;
    }
    return 0;
}

```

La funzione `initCond` calcola il profilo logaritmico in velocità (3.2), in ogni nodo della griglia, aggiungendo una perturbazione casuale. La perturbazione è necessaria affinché si sviluppi la turbolenza. Il parametro `pert` permette di applicare un disturbo pari una determinata percentuale del valore di velocità locale. Il disturbo casuale

viene calcolato a partire dalla funzione `rand` della libreria `std`. Un numero casuale viene generato e diviso per `RAND_MAX`, che è il massimo tra i valori generabili. In questo modo il numero ottenuto è compreso tra 0 e 1. Se il numero generato è dispari viene moltiplicato per  $-1$ . Questo accorgimento permette di applicare perturbazioni negative.

Il parametro  $\rho^+$  nella (3.2) rappresenta la distanza dal bordo del cilindro espressa in coordinate di parete.

Il costruttore della classe `Cylinder`:

```
Cylinder::Cylinder( int argc,
                  char** argv )
:
d( new Private )
```

definisce `d`, che è il puntatore utilizzato per accedere ai membri privati della classe. I parametri della simulazione vengono letti dal file `data` e stampati a video.

Un'altra funzione implementata è la funzione `run` che contiene i comandi eseguiti dal calcolatore quando viene lanciato il file `test_cylinder`.

```
void Cylinder::run()
```

La funzione si occupa della creazione delle classi per la gestione della griglia e del suo partizionamento tra i diversi processori. Si occupa inoltre della costruzione degli spazi a elementi finiti, della costruzione delle matrici e della soluzione del sistema utilizzando i metodi delle classi `FESpace` e `OseenSolver`.

Le linee che seguono si occupano dell'implementazione delle strutture che permettono di imporre condizioni al bordo cicliche per la velocità tra due superfici. È necessario che le due superfici siano identiche per poter creare una corrispondenza univoca tra i rispettivi nodi, per applicare la stessa condizione in velocità.

Vengono mostrate di seguito le linee utilizzate nel caso di dominio cilindrico.

```
//Periodic BC stuff
//construction of the dof
std::map<ID, ID> coupledDofMap;
std::list<std::pair<ID, ID>> coupledFaceList;
{
  std::string refFEString =
    dataFile("fluid/space_discretization/vel_order", "P2");
  const ReferenceFE* refFE(0);
  if ( refFEString.compare("P2") == 0 )
    refFE = &feTetraP2;
  else if ( refFEString.compare("P1") == 0 )
```

```

    refFE = &feTetraP1;
else
    ERROR_MSG(refFEString + "refFE not implemented yet.");

```

Il codice crea la mappa `coupledDofMap` e la lista `coupledFaceList` che sono rispettivamente la mappa utilizzata per l'assegnazione delle condizioni al contorno cicliche e la lista dei nodi accoppiati. Lo spazio di appartenenza della velocità, letto dal file `data`, è immagazzinato nella variabile `refFE`.

```

DOF dof(*fullMeshPtr, *refFE);
DOFInterface3Dto3D dofInterface(*refFE, dof);
dofInterface.update(*fullMeshPtr, INLET, OUTLET,
    AnalyticalSol::isPeriodicCoupleXY, 1e-4);

coupledDofMap = dofInterface.localDofMap();
coupledFaceList = dofInterface.connectedFacetMap();
}
std::cout << "nb Dofs coupled : " << coupledDofMap.size()
    << std::endl;
//end of Periodic BC stuff

```

Le informazioni generate dalle righe precedenti sono utilizzate per creare la tabella `dof` che è necessaria per collegare i gradi di libertà locali con la loro numerazione globale. La corrispondenza tra rispettivi nodi è gestita dall'oggetto `dofInterface`. Il comando `update` genera le connessioni tra le superfici INFLOW e OUTFLOW.

La funzione statica `isPeriodicCoupleXY` controlla che ci sia una corrispondenza tra i nodi delle due superfici, che appartengono al piano `xy`.

Un messaggio stampato a video comunica quanti sono i gradi di libertà accoppiati. Nel caso siano necessarie condizioni cicliche su più di una coppia di superfici, come nel caso della corrente nel canale piano, alcune modifiche devono essere effettuate. È necessaria la creazione di due mappe e due liste per la gestione dei rispettivi accoppiamenti. Una volta che i rispettivi processi di creazione delle connessioni sono stati effettuati, le liste dei nodi accoppiati devono essere fusi in un'unica lista, `coupledFaceList`, che sarà utilizzata al momento di partizionare la griglia fra i diversi processori.

```

std::list<std::pair<ID, ID>>
    coupledFaceList(coupledFaceListYZ);
std::list<std::pair<ID, ID>>
    coupledFaceListTemp(coupledFaceListZX);
coupledFaceList.splice(coupledFaceList.end(),
    coupledFaceListTemp);

```



La lista `coupledFaceList` assicura che i nodi accoppiati siano assegnati allo stesso processore di calcolo.

Per la gestione delle condizioni al bordo vengono creati l'oggetto `bcH` della classe `BCHandler` e alcune funzioni `BCFunctionBase` che possono essere assegnate come condizioni al contorno.

```
// Boundary conditions
BCHandler bcH;

BCFunctionBase uZero( zero_scalar );
BCFunctionBase uConst ( d->getDirConst() );
BCFunctionBase uPois ( d->getU_pois() );

bcH.addBC( "Periodic", INLET, OUTLET,
           Periodic, Full, coupledDofMap, 3);
//bcH.addBC( "Inlet", INLET, Essential, Full, uPois, 3 );
bcH.addBC( "Ringin", RINGIN, Essential, Full, uZero, 3 );
bcH.addBC( "Ringout", RINGOUT, Essential, Full, uZero, 3 );
//bcH.addBC( "Outlet", OUTLET, Natural, Full, uZero, 3 );
bcH.addBC( "Wall", WALL, Essential, Full, uZero, 3 );
```

Le `BCFunctionBase` mostrate sono rispettivamente una funzione omogenea, una funzione costante e il profilo di Poiseuille per il cilindro. Sono disponibili altre funzioni implementate per imporre una determinata portata o profilo. Il comando `addBC` aggiunge una nuova condizione al contorno per il problema.

Nel caso di condizioni non cicliche:

```
bcH.addBC( "Wall", WALL, Essential, Full, uZero, 3 );
```

I parametri passati alla funzione sono:

- il nome della condizione al contorno,
- l'etichetta che identifica la parte di griglia a cui corrisponde la condizione,
- il tipo di condizione al contorno. È possibile applicare oltre alle condizioni di Dirichlet, Essential, e di Neumann, Natural, condizioni di Robin e altre condizioni meno comunemente utilizzate come condizioni di flusso o resistenza,
- il modo in cui la condizione viene applicata: scalare, tangenziale, normale, direzionale e su alcune o tutte le componenti,
- la funzione da applicare,
- il numero di componenti coinvolte.

Nel caso di condizioni cicliche vengono passate le etichette delle superfici accoppiate e la mappa dei gradi di libertà `coupledDofMap`.

```
bcH.addBC( "Periodic", INLET, OUTLET,
           Periodic, Full, coupledDofMap, 3);
```

La griglia può a questo punto essere partizionata tra i vari processori per il calcolo in parallelo, grazie a ParMETIS [10], e vengono creati gli spazi a elementi finiti.

```
//Non periodic problem
// MeshPartitioner< RegionMesh3D<LinearTetra> >
//     meshPart(fullMeshPtr, d->comm);
//Periodic problem
MeshPartitioner<RegionMesh3D<LinearTetra> >
    meshPart(fullMeshPtr, d->comm, 0, 0, &coupledFaceList);

FESpace< RegionMesh3D<LinearTetra>, MapEpetra >
    uFESpace(meshPart, uOrder, 3, d->comm);
FESpace< RegionMesh3D<LinearTetra>, MapEpetra >
    pFESpace(meshPart, pOrder, 1, d->comm);
```

Ora la classe `OseenSolver` può essere costruita con la chiamata al suo costruttore:

```
OseenSolver< RegionMesh3D<LinearTetra> > fluid (oseenData,
                                                uFESpace,
                                                pFESpace,
                                                d->comm, numLM);
```

Il metodo `buildSystem` viene utilizzato per costruire le matrici costanti del sistema. Per calcolare l'energia cinetica del sistema verranno utilizzati i seguenti parametri:

```
//Output file definition to store kinetic energy values
std::ofstream outNorm;
Real kinetic;
vector_Type vel (uFESpace.map(), Repeated);
```

Il vettore `vel` contiene i valori nodali della velocità a un determinato istante temporale, da questi valori si calcola attraverso la norma  $L^2$  il valore dell'energia cinetica, che è salvato in `kinetic`. Il risultato viene copiato tramite l'interfaccia di output `outNorm` nel file di testo `kineticTime`, per ogni passo temporale. L'implementazione di questa procedura è mostrata di seguito.

```
vel.subset(*velAndPressure);
kinetic=uFESpace.l2Norm( vel );
kinetic*=kinetic;
kinetic*=0.5;
```

```

if (verbose){
    outNorm.open("kineticTime", std::ios::app);
    outNorm << time << " " << kinetic << "\n";
    outNorm.close();
}

```

Il parametro `verbose` consente la scrittura a un solo processore. In questo modo viene facilitata la lettura dai risultati nei file di testo ottenuti.

Altri oggetti utilizzati nella costruzione del problema sono:

```

vector_Type velUnique (uFESpace.map(), Unique);

//vector for extrapolated solution
vector_Type beta( fullMap );
//know term vector
vector_Type rhs ( fullMap );
//pointer to the solution vector
vectorPtr_Type velAndPressure (
    new vector_Type(*fluid.solution(), ensight->mapType()));

```

Il vettore `velUnique` viene utilizzato per applicare una condizione iniziale determinata da una funzione. Nel caso del problema di Oseen turbolento il vettore è utilizzato per applicare la (3.2). Il vettore `beta` contiene l'estrapolazione della velocità ottenuta dai valori dei passi temporali precedenti. `rhs` è il vettore del termine noto, mentre `velAndPressure` è un puntatore alla soluzione del problema utilizzato per l'esportazione dei dati.

La soluzione iniziale da applicare al problema viene determinata da una serie di `if`. Il codice confronta il parametro `initial_sol` del file `data` e se trova una corrispondenza applica una determinata condizione iniziale. È possibile applicare una soluzione casuale, un profilo ben definito o la soluzione del problema di Stokes stazionario. È possibile come per le condizioni al contorno applicare un disturbo casuale alla condizione iniziale. Di seguito si mostra il caso in cui la soluzione iniziale scelta è la soluzione del problema di Stokes stazionario:

```

// initialization with stokes solution
if (d->initial_sol == "stokes")
{
    if (verbose) std::cout << std::endl;
    if (verbose) std::cout << "Computing the stokes solution ..."
        << std::endl << std::endl;

    oseenData->dataTime()->setTime(t0);
}

```

```

MPI_Barrier(MPI_COMM_WORLD);

beta *= 0.;
rhs *= 0.;
uFESpace.l2ScalarProduct(d->getSource() , rhs , 0);

fluid.updateSystem(0, beta , rhs );
fluid.iterate( bcH );

//post processing
*velAndPressure = *fluid.solution();
ensight->postProcess( 0 );
fluid.resetPreconditioner();
}

```

La soluzione viene a questo punto inizializzata attraverso la chiamata:

```
bdf.bdfVelocity().setInitialCondition( *fluid.solution() );
```

Le ultime linee di codice mostrate sono quelle utilizzate per la risoluzione del problema di Oseen non stazionario. Il ciclo for:

```

for ( Real time = t0 + dt ;
      time <= tFinal + dt / 2.;
      time += dt , iter++)

```

regola l'avanzamento temporale del problema.

Le costante  $\alpha$  dello schema di avanzamento temporale e l'estrapolazione della velocità  $\beta$  vengono calcolate:

```

double alpha = bdf.bdfVelocity().coefficientFirstDerivative(0)/
               oseenData->dataTime()->timeStep();
beta = bdf.bdfVelocity().extrapolation();

```

La forzante è costruita sommando il valore della funzione source alla matrice di massa per un coefficiente determinato dallo schema di avanzamento temporale:

```

//forcing term
uFESpace.l2ScalarProduct( d->getSource() , rhs , time);
rhs += fluid.matrixMass()*bdf.bdfVelocity()
      .rhsContributionFirstDerivative();

```

Le matrici non costanti del sistema vengono calcolate dal metodo updateSystem. Il sistema costruito viene risolto con un metodo iterativo.

```
fluid.updateSystem( alpha , beta , vel , rhs , d->model );
fluid.iterate( bcH );

bdf.bdfVelocity().shiftRight( *fluid.solution() );
```

La classe `bdf` gestisce lo schema di avanzamento temporale specificatamente per il problema di Navier-Stokes. Il comando `shiftRight` aggiorna i valori del vettore soluzione spostando verso destra quelli corrispondenti a passo temporale precedente.

### 3.5.4 Il file dei dati in ingresso

I dati della simulazione sono inseriti nel file dei dati in ingresso, convenzionalmente chiamato `data`.

Il file è suddiviso in sezioni. Di seguito verranno mostrate quelle rilevanti per le simulazioni con i modelli di turbolenza implementati.

```
[./ problem]
D           = 1.
initial_sol = Log #restart , stokes , none , random , Log
```

La sezione `problem` contiene le informazioni sulla geometria del problema. Il parametro `D` è il diametro del cilindro utilizzato o l'altezza del canale piano e viene utilizzato per calcolare i diversi profili di velocità. La stringa `initial_sol` contiene le informazioni relative alla soluzione iniziale da applicare al problema. È possibile partire dalla soluzione del problema di Stokes stazionario, da condizioni di fluido fermo, da un profilo casuale o da un profilo logaritmico.

Inizializzando la stringa al valore `restart` il programma è in grado di importare i dati di una simulazione effettuata in precedenza. Questa procedura ha presenta però dei problemi. Utilizzando il formato di salvataggio di tipo `ensight` il codice è in grado di importare solamente funzioni appartenenti allo spazio  $\mathcal{P}_1$  poiché `LifeV` salva solamente i valori della soluzione nei vertici dei tetraedri.

```
[./ turbulence]
model           = vanDriest #none , smagorinsky , vanDriest
C-Smagorinsky  = 0.1
flow           = 1.
F              = 1. #forcing term
pert           = 0.05 #random perturbation
```

La sezione `turbulence` contiene i parametri strettamente legati alle novità introdotte nel codice. Il parametro `model` permette di scegliere se applicare o meno un modello

di turbolenza nella risoluzione del problema e il modello da applicare. C-Smagorinsky permette di scegliere il valore della costante  $C_S$  presente nelle (2.14) e (2.15).

La variabile `flow` è utilizzata nel calcolo dei profili di velocità, mentre `F` corrisponde al valore della forzante imposta nella direzione del moto. La variabile `pert` è il valore della perturbazione imposta. La perturbazione è applicata alle condizioni iniziali nel caso in cui vengano applicate condizioni al contorno cicliche, mentre è applicata alla condizione al bordo in ingresso se quest'ultima è di tipo Dirichlet non omogeneo.

```
[../physics]
Re           = 30.
```

La sezione `physics` contiene i valore del numero di Reynolds. Poiché il problema trattato è adimensionalizzato l'unico parametro che è necessario passare al calcolatore è il valore di `Re`.

```
[../time_discretization]
initialtime      = 0.      #start from t_0 + dt
endtime          = 0.1
timestep         = 0.001
BDF_order        = 2

[../space_discretization]
mesh_dir         = ./Mesh/
mesh_file        = rett_prova-t1-5.mesh
mesh_type        = .mesh

vel_order        = P2          # P1, P1Bubble, P2
press_order      = P1          # P1, P2
```

Le sezioni `time_discretization` e `space_discretization` contengono rispettivamente informazioni relative all'avanzamento temporale del problema e alla sua discretizzazione spaziale.

I dati legati al preconditionamento delle matrici e al solutore del sistema algebrico sono rispettivamente nelle sezioni `prec` e `solver`.

Un'altra sezione importante è quella che tratta la stabilizzazione. `LifeV` contiene gli strumenti per applicare una stabilizzazione *Interior Penalty* [4]. I valori dei coefficienti del metodo IP sono nella sezione `ipstab`. Poiché si sviluppi la turbolenza è però necessario non applicare nessuna stabilizzazione ulteriore a quella introdotta dal modello di Smagorinsky. Per questo motivo i coefficienti della sezione `ipstab` devono essere tutti pari a zero:

```
[../ipstab]
```

```

gammaBeta      = 0.    #0.125
gammaDiv       = 0.    #0.125
gammaPress     = 0.    #0.03125

```

La parte di codice mostrata di seguito è strettamente legata alle prestazioni.

```

[../ solver]
solver         = gmres
conv           = rhs
max_iter       = 150
reuse          = true
max_iter_reuse = 80
kspace         = 100
tol            = 1.e-10 # AztecOO tolerance

```

Il sistema lineare viene risolto tramite un metodo iterativo. In questa sezione l'utente può scegliere i parametri relativi al metodo. `solver` è il tipo di metodo scelto, `conv` è il parametro utilizzato per la normalizzazione del residuo, `max_iter` il numero massimo di iterazioni per raggiungere la convergenza, `kspace` la dimensione dello spazio di Krylov che viene utilizzato per approssimare la soluzione e `tol` la tolleranza scelta.

Il parametro `reuse` indica se il preconditionatore deve essere ricalcolato anche in caso di convergenza del metodo. Se `reuse=true` e la convergenza è raggiunta in un numero di iterazioni maggiore di `max_iter_reuse` il preconditionatore verrà ricalcolato al passo temporale successivo, altrimenti no. Gli ultimi due parametri descritti sono di fondamentale importanza per le prestazioni del codice.

Si presenta una breve analisi dei tempi di calcolo delle simulazioni.

In Tabella 3.1 vengono indicati il numero di vertici e tetraedri delle griglie utilizzate, lo spazio a elementi finiti a cui appartiene la velocità e la presenza o meno del modello di turbolenza. Le ultime due righe contengono i tempi medi di calcolo per il singolo passo temporale:  $T_{precond}$  nel caso in cui il preconditionatore venga ricalcolato,  $T_{non\ precond}$  nel caso in cui venga riutilizzato il preconditionatore

Vertici	20384	20384	20384	16650	16650	1440	1440
Tetraedri	109350	109350	109350	89784	89784	6534	6534
Spazio $\mathbf{u}$	$\mathcal{P}_2$	$\mathcal{P}_2$	$\mathcal{P}_{1b}$	$\mathcal{P}_2$	$\mathcal{P}_2$	$\mathcal{P}_2$	$\mathcal{P}_{1b}$
Modello turbolenza	no	sì	sì	no	sì	no	no
T precond.	155s	300s	38s	100s	155s	45s	11s
T non precond.	70s	100s	20s	50s	50s	5s	5s

Tabella 3.1: Tempi medi per iterazione.

calcolato nei passi precedenti.

Come mostrato in Tabella 3.1, il calcolo del preconditionatore è l'operazione più onerosa nel corso della simulazione. Per questo motivo il numero di passi temporali risolti dalle simulazioni effettuate sulle griglie più raffinate, con  $\mathbf{u} \in \mathcal{P}_2$ , varia da un minimo di 250 a un massimo di circa 1000 al giorno. Il valore più basso si ottiene con `reuse=false`, ricalcolando quindi il preconditionatore ad ogni passo temporale.

La scelta ottimale dei parametri `reuse` e `max_iter_reuse` permetterà tempi di calcolo minori.

Nell'ambito della simulazione numerica della turbolenza i tempi di calcolo ottenuti risultano molto elevati. La turbolenza richiede infatti che il passo temporale sia sufficientemente piccolo per essere confrontabile con le scale temporali turbolente, ma richiede un numero elevato di iterazioni per raggiungere il regime, dell'ordine di  $10^4$ .

Una parziale soluzione del problema sarebbe quella di passare a una trattazione esplicita del termine convettivo, diminuendo così il numero di operazioni da compiere ad ogni passo temporale. Il problema in forma esplicita deve però soddisfare la condizione CFL [16] che limita il passo temporale al di sotto di certi valori. Per alcune configurazioni dei parametri,  $\Delta t$  può risultare molto piccolo e quindi aumentare il costo computazionale della simulazione.

La soluzione ideale è quindi quella di ottimizzare il codice per migliorarne le prestazioni.



## Geometrie e parametri del problema

In questo capitolo vengono descritti i domini e alcuni parametri utilizzati nelle simulazioni. Ci si è concentrati sullo studio del problema del flusso in un canale cilindrico e della corrente nel canale piano.

Vengono mostrate di seguito le caratteristiche delle griglie di calcolo e le condizioni al contorno utilizzate per i differenti problemi. Le griglie sono costituite da elementi tetraedrici.

Viene presentata inoltre la relazione di conversione tra  $Re$  e  $Re_\tau$  per il problema del flusso nel canale cilindrico a partire da dati presenti in letteratura.

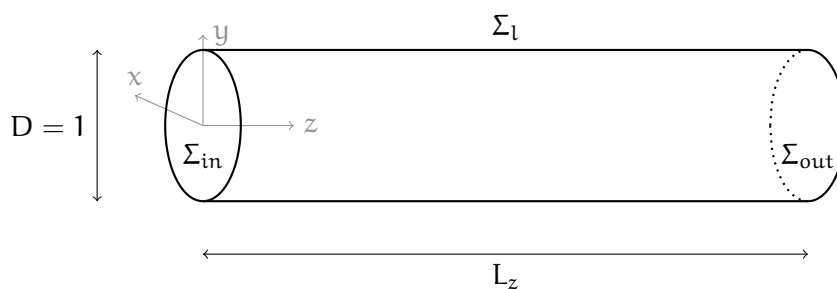
La scelta di geometrie semplici è determinata dalla necessità di validare il modello implementato confrontandosi con casi test ben documentati in letteratura.

### 4.1 Flusso nel canale cilindrico

Il primo problema affrontato è quello di un flusso in un tubo cilindrico.

La Figura 4.1, mostra il canale cilindrico utilizzato nelle simulazioni.

La lunghezza del dominio in direzione  $z$  è variabile,  $D = 1$  è il diametro del cilindro.



**Figura 4.1:** Dominio di calcolo, canale cilindrico.

Per le simulazioni su dominio cilindrico si applica una condizione di Dirichlet omogenea sulla superficie laterale  $\Sigma_l$ , mentre sono state utilizzate diverse condizioni per le superfici di ingresso e uscita.

Nell'ambito delle simulazioni in regime turbolento le condizioni più comunemente utilizzate sono di tipo ciclico per la velocità tra le facce  $\Sigma_{in}$  e  $\Sigma_{out}$ , tubo esteso infinitamente. In questo caso una forzante, il cui valore verrà calcolato nella prossima sezione, deve essere applicata perché si sviluppi il moto nel cilindro.

Un approccio differente al problema è quello di generare il moto applicando una portata alla superficie di ingresso. Questo viene fatto con una condizione di Dirichlet non omogenea sulla parete in ingresso e una condizione di sforzo normale nullo, condizione di Neumann omogenea, sulla superficie di uscita. La condizione d'uscita corrisponde a quella di un tubo che sfocia all'aperto.

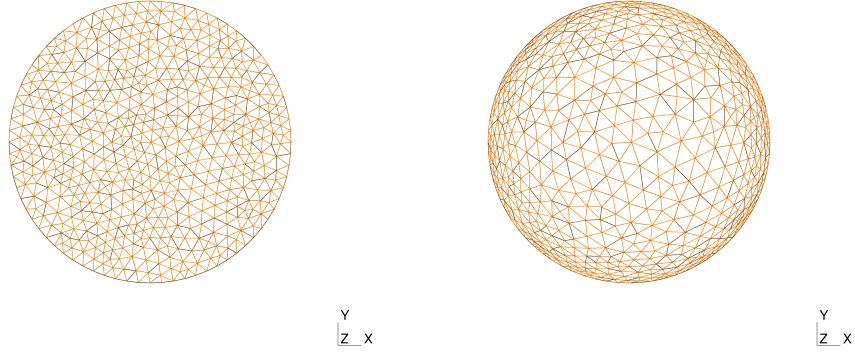
Specialmente per  $Re$  elevati, è necessario raffinare la griglia vicino alla superficie laterale del cilindro in modo da garantire un numero sufficiente di nodi per la risoluzione dello strato limite. Questo procedimento viene effettuato in coordinate cilindriche  $(\rho, \theta)$  sul piano  $xy$ :

$$\begin{aligned} & x_v, y_v, \\ & \rho_v = \sqrt{x^2 + y^2}, \\ & \theta = \arccos\left(\frac{x_v}{\rho_v}\right). \\ & \Downarrow \\ & \rho_r = R \frac{\tanh(k\rho_v)}{\tanh(kR)}, \\ & x_r = \rho_r \cos(\theta), \\ & y_r = \rho_r \sin(\theta). \end{aligned}$$

Le variabili  $x_v$  e  $y_v$  sono le coordinate della griglia non raffinata. Calcolata coordinata  $\rho_v$ , distanza dal centro, è possibile risalire al valore della coordinata  $\theta$ . Applicando la tangente iperbolica a  $\rho_v$  si ottiene la coordinata raffinata  $\rho_r$ . Il denominatore della frazione ha il solo scopo di mantenere costanti le dimensioni del cilindro. Infine vengono calcolate le coordinate della griglia raffinata  $x_r$  e  $y_r$ . Il parametro  $R$  è il raggio del cilindro.

Un esempio è mostrato in Figura 4.2.

Per il problema del flusso in un tubo cilindrico il valore del numero di Reynolds critico  $Re_c = \frac{UD}{\nu}$ , basato sulla velocità media  $U$ , è circa 2300.



**Figura 4.2:** Canale cilindrico, griglia non raffinata a sinistra, griglia raffinata con  $\tanh(3.5\rho)$  a destra. Piano  $xy$ .

#### 4.1.1 Calcolo della forzante e fattore di conversione

Nel caso di un'adimensionalizzazione basata sulla velocità d'attrito  $V_*$ , grazie ai dati presenti in [18], è stato possibile calcolare analiticamente il valore della forzante  $f$  da applicare in direzione  $z$  per generare il moto.

Per un dominio cilindrico è possibile esprimere lo sforzo a parete come

$$\tau_w = \frac{\Delta p R}{L_z 2}, \quad (4.1)$$

dove  $\frac{\Delta p}{L_z}$  è il valore della forzante cercato e  $R$  il raggio del cilindro.

Dalla definizione della velocità d'attrito a parete (1.6) è possibile ricavare l'equazione per lo sforzo a parete

$$\tau_w = \rho V_*^2, \quad (4.2)$$

con  $\rho$  densità del fluido.

Dal confronto della (4.2) e della (4.1) si ottiene

$$\rho V_*^2 = \frac{\Delta p R}{L_z 2} \Rightarrow \frac{\Delta p}{L_z} = 4 \frac{V_*^2}{D} \rho. \quad (4.3)$$

Poiché nelle simulazioni effettuate  $D = 1$ ,  $\rho = 1$  e  $V_* = 1$ , sostituendo questi valori nella (4.3), si trova il valore di  $f_z$

$$f_z = \frac{\Delta p}{L_z} = 4. \quad (4.4)$$

Da [18] si ricavano inoltre le informazioni necessarie per passare da  $Re$  basato sulla velocità media  $U$ , al corrispondente valore di  $Re_\tau$ . Il gradiente di pressione può

essere infatti espresso come

$$\frac{\Delta p}{L_z} = \frac{\lambda}{2D} \lambda \rho U^2,$$

in cui  $\lambda$  è il coefficiente di resistenza adimensionale e  $U$  la velocità media del fluido.

La (4.2) diventa quindi

$$\tau_w = \frac{1}{8} \lambda \rho U^2. \quad (4.5)$$

Grazie a una serie di esperimenti, effettuati da Blasius, è stato possibile stabilire la seguente relazione empirica per calcolare il valore di  $\lambda$

$$\lambda = 0.3164 \left( \frac{UD}{\nu} \right)^{-1/4} = 0.3164 \text{Re}^{-1/4}, \quad (4.6)$$

dove  $\nu$  è la viscosità cinematica del fluido.

Partendo dalla definizione di  $\text{Re}_\tau$  (1.7), utilizzando (1.5) (1.6), (4.5) e (4.6) si ottiene:

$$\begin{aligned} \text{Re}_\tau &= \frac{V_* L}{\nu} = \frac{\sqrt{\tau_w / \rho} D}{\nu} = \frac{\sqrt{\frac{0.3164}{8} \text{Re}^{-1/4} U^2 D}}{\nu} \\ &= \sqrt{\frac{0.3164}{8}} \text{Re}^{-1/8} \frac{UL}{\nu} = \sqrt{\frac{0.3164}{8}} \text{Re}^{7/8}. \end{aligned} \quad (4.7)$$

## 4.2 Corrente nel canale piano

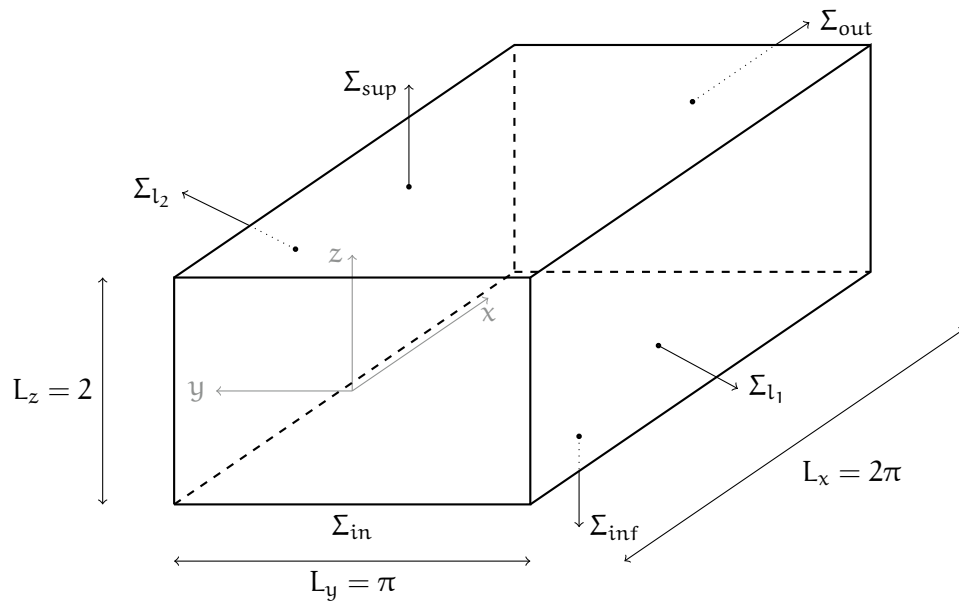
Un altro problema affrontato in questo lavoro è quello relativo al moto di un fluido tra due lastre piane parallele infinite, Figura 4.3. Le dimensioni del dominio sono  $2\pi \times \pi \times 2$ .

Le superfici  $\Sigma_{\text{inf}}$  e  $\Sigma_{\text{sup}}$ , che rappresentano le due lastre piane, sono considerate ferme. Ad esse viene applicata una condizione di Dirichlet omogenea per la velocità, che corrisponde a una condizione di non scorrimento a parete.

Le condizioni cicliche per la velocità applicate tra le superfici  $\Sigma_{l_1}$ ,  $\Sigma_{l_2}$  e tra le superfici  $\Sigma_{\text{in}}$ ,  $\Sigma_{\text{out}}$  approssimano la condizione di canale illimitato nelle direzioni  $x$  e  $y$ .

Il moto all'interno del canale viene generato dall'applicazione di una forzante in direzione  $x$  di valore  $\frac{2}{\text{Re}}$  nel caso si utilizzi un numero di Reynolds basato sulla velocità a centro canale. Nel caso in cui il numero di Reynolds si quello basato sulla velocità d'attrito il valore della forzante è pari a 1.

Poiché inizialmente le condizioni cicliche non erano ancora implementate in LifeV, in una prima fase del lavoro sono state utilizzate diverse combinazioni di condizioni



**Figura 4.3:** Dominio di calcolo, canale piano.

al contorno che meglio vi si avvicinassero.

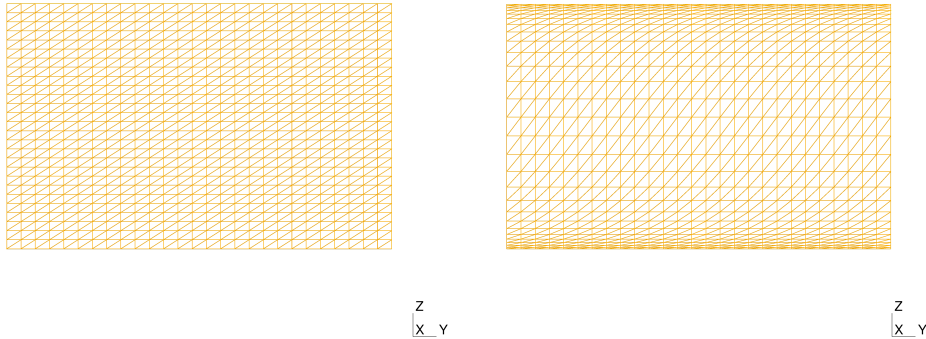
In questa fase, alle facce  $\Sigma_{l_1}$ ,  $\Sigma_{l_2}$ ,  $\Sigma_{in}$  e  $\Sigma_{out}$  sono state applicate condizioni di Neumann di sforzo normale nullo, per rappresentare la situazione di una corrente sviluppata. Tuttavia si è verificato che per numeri di Reynolds elevati si notava un importante flusso trasversale vicino alla superficie di ingresso. Questo fenomeno potrebbe essere dovuto a dei gradienti di pressione trasversali generati per effetto delle condizioni al bordo di Neumann sulle superfici laterali.

Anziché imporre una forzante, è possibile generare il moto nel canale imponendo la portata. Questo si ottiene mediante una condizione di Dirichlet non omogenea applicata alla superficie  $\Sigma_{in}$  insieme a una condizione di Neumann omogenea per le altre tre superfici.

La griglia costruita sul dominio in Figura 4.3 ha spaziatura uniforme in direzione  $x$  e  $y$ . In direzione  $z$  la griglia è stata raffinata con la funzione  $\tanh(kz)$ . Un esempio è mostrato in Figura 4.4. La variabile  $z$  rappresenta la coordinata del punto, mentre il parametro  $k$  è responsabile di un maggiore o minore raffinamento vicino alle pareti superiore e inferiore.

Il raffinamento è necessario al fine di garantire un sufficiente numero di nodi nello strato limite. In questo modo il moto del fluido nello strato limite può essere risolto correttamente.

Il parametro  $k$  è dipendente dal valore di  $Re_\tau$ . All'aumentare del numero di Reynolds



**Figura 4.4:** Canale piano, griglia non raffinata a sinistra, griglia raffinata con  $\tanh(4z)$  a destra. Piano  $yz$ .

infatti si avrà una riduzione dello spessore dello strato limite. Il valore di  $k$  dovrà quindi essere incrementato per garantire un maggior raffinamento al bordo. Sarà inoltre necessario aumentare il numero dei punti della griglia per evitare che sia troppo lasca a centro canale.

Il valore critico del numero di Reynolds basato sulla velocità a centro canale per questo problema è  $Re_c \simeq 2935$ .

In regime turbolento è possibile esprimere il profilo di velocità longitudinale medio in funzione della distanza dalla parete.

Nella zona prossima alla parete, detta sottostrato viscoso, gli sforzi turbolenti sono trascurabili e esiste una relazione lineare tra il valore della velocità e la distanza a parete  $z^+$  (1.8)

$$u_x = z^+ \quad z^+ < 5.$$

Per  $5 < z^+ < 30$  ci si trova nella cosiddetta zona di buffer, in cui la velocità è una funzione di  $z^+$

$$u_x = V_* f(z^+) \quad 5 < z^+ < 30. \quad (4.8)$$

La funzione che descrive la velocità nella zona di buffer deve raccordare il sottostrato viscoso con la regione logaritmica,  $z^+ > 30$ , in cui vale la seguente relazione

$$u_x = 2.5 \log z^+ + 5.5 \quad z^+ > 30.$$

Il profilo medio di velocità applicato alle simulazioni, ricavato da [18] e [13], è:

$$\begin{cases} \mathbf{u}_x = 2.5 \log z^+ + 5.5 & \text{se } z^+ > 11.6, \\ \mathbf{u}_x = z^+ & \text{se } z^+ \leq 11.6, \\ \mathbf{u}_y = 0, \\ \mathbf{u}_z = 0. \end{cases} \quad (4.9)$$

Nella regione di buffer per  $z^+ = 11.6$  avviene il raccordo tra le curve caratteristiche del sottostrato viscoso e della regione logaritmica.

## Risultati: regime laminare

Le prime simulazioni presentate in regime di corrente laminare hanno lo scopo di verificare l'effetto dissipativo del modello implementato. I valori scelti per il numero di Reynolds sono infatti molto minori dei valori critici.

Per prima cosa sono stati scelti dei valori del parametro  $Re$ , calcolato rispetto alla velocità a centro canale, per cui le simulazioni effettuate senza modelli di turbolenza hanno mostrato un regime laminare.

Di seguito vengono presentati i risultati ottenuti per i due domini descritti nel Capitolo 4.

### 5.1 Flusso nel canale cilindrico: $Re_\tau = 90$

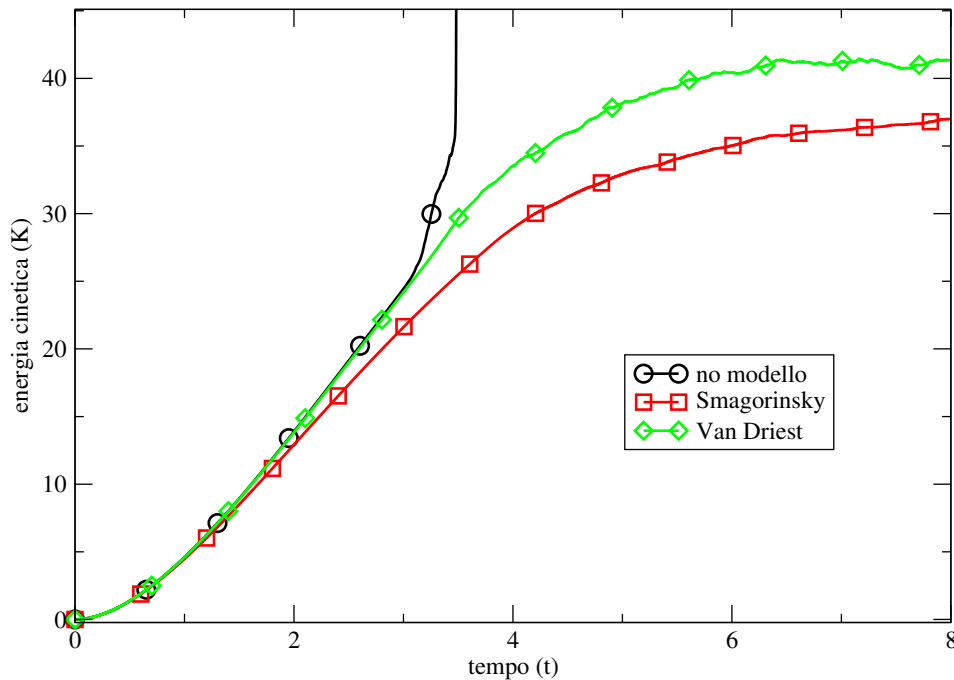
I risultati presentati sono relativi a un caso test nel dominio cilindrico, Figura 4.1, con un valore del numero di Reynolds turbolento  $Re_\tau = 90$ . Questo valore corrisponde a un numero di Reynolds, calcolato rispetto alla velocità media (4.7),  $Re \simeq 1084$  in condizione laminare e ben lontano dal valore critico.

La simulazione è stata effettuata su una griglia cilindrica di diametro  $D = 1$  e lunghezza  $L_z = 5$ , costruita con il programma gmhs [6]. La griglia è composta da 1050 vertici e 4680 tetraedri. La singola superficie circolare è ripetuta 20 volte e contiene 20 nodi sulla sua circonferenza. Non è necessario introdurre alcun raffinamento per la griglia poiché il numero di nodi nello strato limite è sufficiente alla sua buona risoluzione.

Sono applicate condizioni di Dirichlet omogenee su  $\Sigma_1$  e condizioni cicliche sulla velocità tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .

Le condizioni iniziali sono di fluido fermo all'interno del condotto,  $\mathbf{u}_0 = \mathbf{0}$ . Per generare il moto viene applicata una forzante in direzione  $z$  pari a  $\mathbf{f}_z = 4$ .





**Figura 5.1:** Canale cilindrico  $Re_\tau = 90$ . Energia cinetica in funzione del tempo. Confronto fra i diversi modelli di turbolenza.

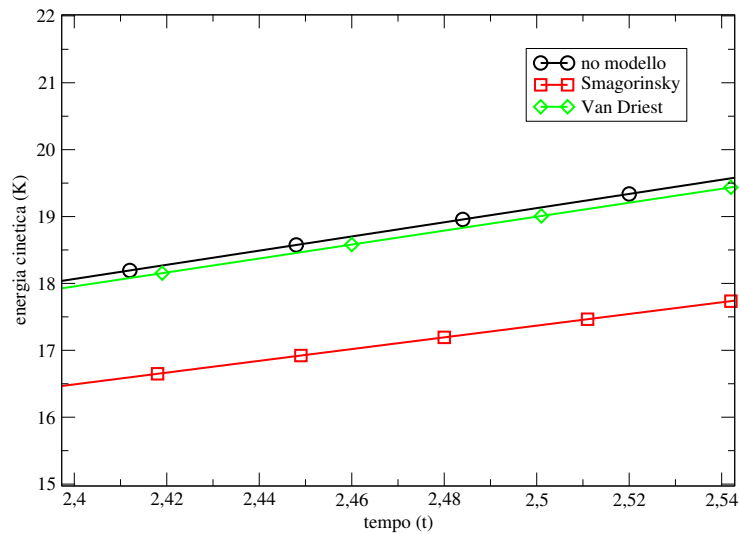
Il modello di Smagorinsky utilizza un valore della costante  $C_S = 0.1$  e l'ampiezza del filtro viene calcolata come la radice cubica del volume dell'elemento locale.

Il passo temporale utilizzato è  $\Delta t = 0.001$ , mentre il tempo finale  $T = 10$  è stato scelto per permettere alle simulazioni di raggiungere il regime.

Lo schema di avanzamento temporale è lo schema *Backward Difference* di secondo ordine. La coppia velocità-pressione  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ .

Osservando le tre curve in Figura 5.1 si nota un andamento crescente dell'energia cinetica all'aumentare del tempo. Questo è in accordo con le condizioni iniziali applicate: il flusso all'interno del cilindro si sta infatti sviluppando per effetto della forzante.

L'effetto della viscosità introdotta dai modelli è inoltre confermato dalle rispettive posizioni delle tre curve. Come si può notare in Figura 5.2, la curva senza alcun modello descrive un'energia maggiore rispetto alle altre due, questo perché nessuna viscosità turbolenta è introdotta. Inoltre si può notare come la curva in verde, corrispondente al modello con la correzione di Van Driest, sia più vicina alla curva nera rispetto alla curva rossa, modello di Smagorinsky. Questo accade perché la correzione di Van Driest smorza la quantità di viscosità numerica introdotta al diminuire



**Figura 5.2:** Canale cilindrico  $Re_\tau = 90$ . Energia cinetica in funzione del tempo. Confronto fra i diversi modelli di turbolenza, dettaglio.

della distanza dalla parete.

L'intensità ridotta delle differenze tra le tre curve è dovuta al fatto che la viscosità turbolenta introdotta dai modelli è piccola rispetto alla viscosità cinematica.

Nonostante il comportamento qualitativo dei modelli risulti corretto, i risultati ottenuti non sono soddisfacenti. Come mostra la Figura 5.1 la simulazione senza modello di turbolenza smette di convergere dopo alcune iterazioni. Inoltre è possibile notare delle oscillazioni dell'energia cinetica sia nella curva in verde che, seppur con minore intensità, nella curva in rosso.

I profili di velocità si dimostrano fortemente non stazionari e instabili.

Questo andamento è però in disaccordo con la fisica del problema. Il valore del numero di Reynolds calcolato rispetto alla velocità a centro canale, alla comparsa delle prime instabilità, è  $Re \simeq 1000$ . Questo valore colloca la simulazione all'interno del regime laminare dove il profilo dovrebbe avere un andamento parabolico senza oscillazioni.

Le instabilità trovate potrebbero essere dovute allo spazio a elementi finiti utilizzato per la velocità. Lo spazio  $\mathcal{P}_2$  è infatti meno dissipativo della spazio  $\mathcal{P}_{1b}$ . Un confronto tra i risultati ottenuti con i due differenti spazi è mostrato per il problema relativo alla corrente nel canale piano.

## 5.2 Corrente nel canale piano: $Re_\tau = 30$

Il valore scelto per il numero di Reynolds turbolento,  $Re_\tau = 30$ , è ben lontano dai valori,  $Re_\tau \geq 180$ , del regime turbolento. Lo scopo della simulazione, che avviene quindi in regime laminare, è di verificare l'effetto della viscosità turbolenta introdotta con i modelli implementati.

In questa simulazione sono inoltre applicate le condizioni iniziali corrispondenti al profilo di velocità logaritmico turbolento (4.9).

Le dimensioni del dominio sono, come mostrato in Figura 4.3,  $2\pi \times \pi \times 2$ .

La griglia costruita con il programma gmhs [6] ha  $10 \times 12 \times 12$  nodi, per un totale di 1440 vertici e 6534 tetraedri. È stato effettuato un raffinamento lungo la coordinata  $z$  con la funzione  $\tanh(1.5z)$ .

Sono applicate condizioni di Dirichlet omogenee su  $\Sigma_{inf}$  e  $\Sigma_{sup}$  e condizioni cicliche sulla velocità tra  $\Sigma_{in}$ ,  $\Sigma_{out}$  e tra  $\Sigma_{l_1}$  e  $\Sigma_{l_2}$ .

Vengono applicate le condizioni iniziali (4.9) e una forzante in direzione  $z$  pari a  $f_z = 1$  per mantenere il moto all'interno del canale.

Il modello di Smagorinsky utilizza un valore della costante  $C_S = 0.1$  e l'ampiezza del filtro viene calcolata come la radice cubica del volume dell'elemento locale.

Il passo temporale utilizzato è  $\Delta t = 0.001$ , mentre il tempo finale  $T = 10$ .

Lo schema di avanzamento temporale è lo schema *Backward Difference* di secondo ordine. La velocità  $\mathbf{u}$  appartiene allo spazio a elementi finiti  $\mathcal{P}_2$ , mentre la pressione  $p$  allo spazio  $\mathcal{P}_1$ .

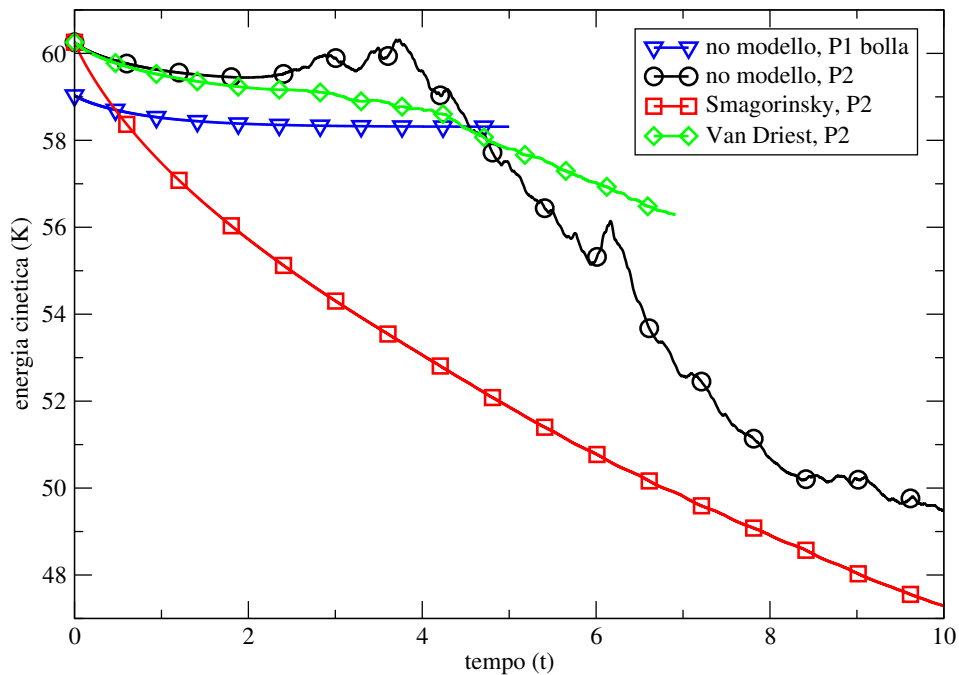
La Figura 5.3 mostra un andamento decrescente dell'energia cinetica in funzione dell'avanzamento temporale. Questo accade per effetto delle condizioni iniziali applicate. Poiché queste condizioni corrispondono a quelle di un profilo turbolento sviluppato, mentre il regime è laminare, la velocità tende verso un profilo differente da quello imposto inizialmente.

La Figura 5.3 mostra inoltre come i valori dell'energia cinetica siano superiori per la simulazione senza modello di turbolenza, come ci si dovrebbe aspettare.

Il grafico mostra inoltre un andamento decrescente molto più marcato per la curva relativa al modello di Smagorinsky, in rosso, effetto diretto della sovradiffusione che lo caratterizza. Il fattore di smorzamento introdotto dalla correzione di Van Driest (1.15) fa sì che la curva in verde segua più fedelmente l'andamento della simulazione senza modello, in nero.

Come nel caso del cilindro la simulazione presenta delle instabilità.

La Figura 5.3 mostra per  $t > 2$  delle oscillazioni dell'energia cinetica. In corrispon-



**Figura 5.3:** Canale piano  $Re_\tau = 30$ . Energia cinetica in funzione del tempo. Confronto fra i diversi modelli di turbolenza.

denza di queste oscillazioni si notano le prime instabilità nel profilo di velocità, che aumentano di intensità nel corso della simulazione.

Le instabilità riscontrate sono più forti nel caso senza modello di turbolenza. Anche i valori di velocità raggiunti a centro canale sono più elevati per la simulazione senza modello.

Il valore del numero di Reynolds rispetto alla velocità a centro canale calcolato nell'istante in cui si mostrano le prime instabilità è  $Re \simeq 400$ . Tale valore aumenta fino a un massimo  $Re \simeq 600$  nella simulazione senza modello di turbolenza. Entrambi i valori sono al di sotto del valore critico.

L'ipotesi è che queste instabilità siano dovute allo spazio a elementi finiti  $\mathcal{P}_2$  utilizzato per la discretizzazione spaziale della velocità. A sostegno di questa ipotesi è stata effettuata una simulazione con gli stessi parametri delle curve in nero, rosso e verde, con l'unica differenza che la velocità  $\mathbf{u} \in \mathcal{P}_{1b}$ . La curva in blu, Figura 5.3, mostra l'andamento dell'energia cinetica per questa simulazione. La simulazione raggiunge il regime e nessuna instabilità si manifesta.

Il motivo di queste instabilità sembra quindi dovuto al minor potere dissipativo dello spazio  $\mathcal{P}_2$  rispetto allo spazio  $\mathcal{P}_{1b}$ .

### 5.2.1 Soluzione analitica

I problemi riscontrati in regime laminare hanno mostrato l'esigenza di controllare l'effetto che il termine di Smagorinsky ha sul problema risolto.

Di conseguenza si è effettuato un confronto tra la soluzione numerica ottenuta a regime e la soluzione analitica delle equazioni di Oseen.

Il problema di Oseen adimensionalizzato a cui viene aggiunto il termine di Smagorinsky è

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \boldsymbol{\beta} \cdot \nabla \mathbf{u} + \nabla p - \frac{1}{Re_\tau} \Delta \mathbf{u} - \nabla \cdot \left( 2C_S^2 \bar{\Delta}^2 |S| S(\mathbf{u}) \right) = 0 & \text{in } \Omega, \\ \mathbf{u} = 0 & \text{su } \partial\Omega. \end{cases} \quad (5.1)$$

Si cercano soluzioni stazionarie del problema (5.1) in cui la velocità è nulla in direzione  $y$  e  $z$  ed è della sola coordinata verticale, mentre la pressione è funzione della coordinata  $x$ :

$$u_x = u_x(z), \quad u_y = u_z = 0, \quad p = p(x).$$

Queste condizioni su velocità e pressione sono caratteristiche del problema della corrente in un canale piano in regime laminare.

Il risultato cercato dovrà essere diverso dal profilo di Poiseuille a causa dell'introduzione del termine di Smagorinsky.

Sotto le ipotesi di stazionarietà e sfruttando le particolari proprietà della velocità, i primi due termini della (5.1) possono essere eliminati.

Poiché le componenti  $y$  e  $z$  della velocità sono nulle è possibile ridurre l'equazione vettoriale (5.1) solo alla sua prima componente, che corrisponde ad una equazione scalare nelle sole incognite  $u_x$  e  $p$

$$\frac{\partial p}{\partial x} = \frac{1}{Re_\tau} \frac{\partial^2 u_x}{\partial z^2} + \frac{\partial}{\partial z} \left[ 2C_S^2 \bar{\Delta}^2 \frac{\partial u_x}{\partial z} \text{sign} \left( \frac{\partial u_x}{\partial z} \right) \frac{1}{2} \frac{\partial u_x}{\partial z} \right]. \quad (5.2)$$

La funzione  $\text{sign}(\cdot)$  è la funzione segno. Per semplificare la notazione si definisce la costante  $\alpha = 2C_S^2 \bar{\Delta}^2$ .

Osservando la (5.2) si nota che la derivata a sinistra dell'uguale dipende dalla sola coordinata  $x$ , mentre le derivate a destra dell'uguale dipendono solamente dalla coordinata  $z$ . Perché l'uguaglianza sia valida le derivate devono essere uguali alla medesima costante  $k$ . La (5.2) può quindi essere riscritta come

$$k = \frac{\partial p}{\partial x} = \frac{1}{Re_\tau} \frac{\partial^2 u_x}{\partial z^2} + \alpha \frac{\partial}{\partial z} \left[ \frac{1}{2} \left( \frac{\partial u_x}{\partial z} \right)^2 \text{sign} \left( \frac{\partial u_x}{\partial z} \right) \right]. \quad (5.3)$$

La costante  $k$  è quindi equivalente al gradiente di pressione rispetto alla coordinata longitudinale. Questa quantità è uguale, per il problema della corrente nel canale

piano adimensionalizzato rispetto a  $V_*$ , alla forzante  $f_x$  cambiata di segno. Il valore di  $k$  è quindi  $-1$ . Dopo alcuni passaggi matematici, rinominando  $\xi = \frac{\partial u_x}{\partial z}$  si ottiene

$$\frac{\partial \xi}{\partial z} \left( \frac{1}{Re_\tau} + \alpha \xi \text{sign}(\xi) \right) = k = -1.$$

Separando le variabili  $\xi$  e  $z$  e integrando, si ottiene l'equazione di secondo grado nella variabile  $\xi$

$$\xi^2 + \frac{2}{\alpha Re_\tau} \xi \text{sign}(\xi) - \frac{2}{\alpha} (-z + c_1) \text{sign}(\xi) = 0, \quad (5.4)$$

dove  $c_1$  è la costante di integrazione. Le soluzioni di (5.4) sono

$$\xi_{1,2} = -\frac{1}{\alpha Re_\tau} \text{sign}(\xi) \pm \sqrt{\frac{1}{\alpha^2 Re_\tau^2} + \frac{2}{\alpha} (-z + c_1) \text{sign}(\xi)}. \quad (5.5)$$

La soluzione (5.5) può essere scritta in forma estesa grazie ad alcune considerazioni di tipo algebrico, sfruttando la dipendenza di  $\xi_{1,2}$  dal suo stesso segno.

Se  $\xi > 0$  allora

$$-\frac{1}{\alpha Re_\tau} \pm \sqrt{\frac{1}{\alpha^2 Re_\tau^2} + \frac{2}{\alpha} (-z + c_1)} > 0.$$

La soluzione con il segno meno è scartata per ovvi motivi. Rimane quindi la soluzione

$$-\frac{1}{\alpha Re_\tau} + \sqrt{\frac{1}{\alpha^2 Re_\tau^2} + \frac{2}{\alpha} (-z + c_1)} > 0,$$

che è verificata per

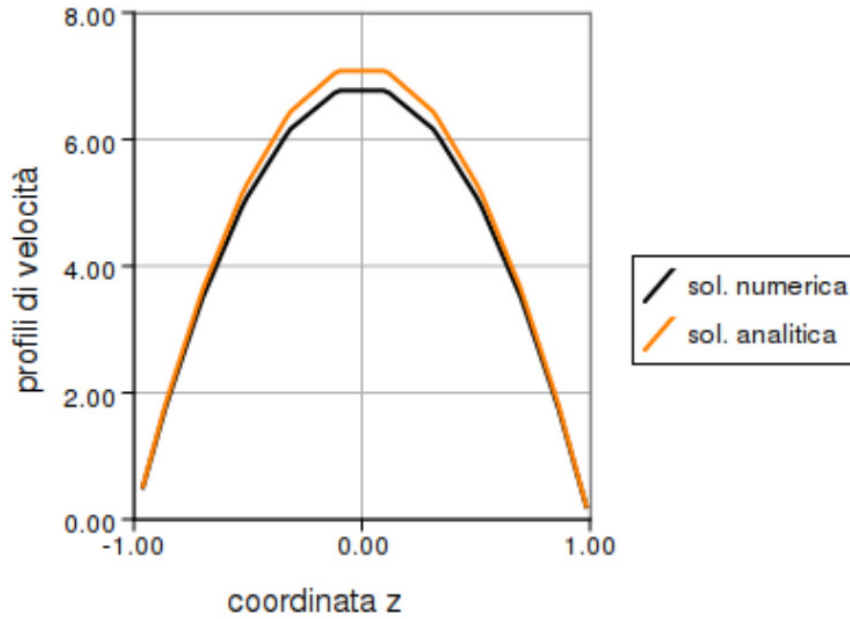
$$\frac{2}{\alpha} (-z + c_1) > 0 \quad \Rightarrow \quad z < c_1.$$

Con considerazioni analoghe per  $\xi < 0$  è possibile riscrivere (5.5) come

$$\xi = \begin{cases} -\frac{1}{\alpha Re_\tau} + \sqrt{\frac{1}{\alpha^2 Re_\tau^2} + \frac{2}{\alpha} (-z + c_1)} & \text{se } z < c_1, \\ \frac{1}{\alpha Re_\tau} - \sqrt{\frac{1}{\alpha^2 Re_\tau^2} - \frac{2}{\alpha} (-z + c_1)} & \text{se } z \geq c_1. \end{cases}$$

Per simmetria e regolarità la velocità a centro canale deve soddisfare  $\xi(0) = 0$ . Questa condizione è verificata per un valore della costante  $c_1 = 0$ . Il risultato ottenuto con opportuni raccoglimenti è

$$\xi = \begin{cases} \frac{1}{\alpha Re_\tau} (-1 + \sqrt{1 - \eta z}) & \text{se } z < 0, \\ \frac{1}{\alpha Re_\tau} (1 - \sqrt{1 + \eta z}) & \text{se } z \geq 0, \end{cases} \quad (5.6)$$



**Figura 5.4:** Canale piano  $Re_\tau = 15$ . Confronto tra soluzione numerica e analitica. Profili di velocità in direzione  $x$ .

dove si è posto  $\eta = 2Re_\tau^2 \alpha$  per semplificare la notazione.

Per trovare la soluzione analitica  $u_x$  della (5.1) è sufficiente integrare la (5.6) rispetto alla variabile  $z$ . Il risultato è la seguente funzione pari

$$u_x = \begin{cases} \frac{1}{\alpha Re_\tau} \left[ -z - \frac{2}{3\eta} (1 - \eta z)^{3/2} \right] + c_2 & \text{se } z < 0, \\ \frac{1}{\alpha Re_\tau} \left[ z - \frac{2}{3\eta} (1 + \eta z)^{3/2} \right] + c_2 & \text{se } z \geq 0. \end{cases} \quad (5.7)$$

Le condizioni al contorno  $u_x(-1) = u_x(1) = 0$  forniscono il valore per la costante di integrazione  $c_2$

$$c_2 = -\frac{1}{\alpha Re_\tau} \left[ 1 - \frac{2}{3\eta} (1 + \eta)^{3/2} \right].$$

I parametri  $Re_\tau$  e  $\alpha$  presenti nella (5.7) sono ricavati a partire dai dati della simulazione.

$Re_\tau$  è il numero di Reynolds basato sulla velocità d'attrito utilizzato nella simulazione. Per calcolare il valore di  $\alpha$  e  $\eta$  sono necessari il valore della costante di Smagorinsky  $C_s$ , decisa a priori dall'utente, e l'ampiezza del filtro  $\bar{\Delta}$  per la quale è

stato utilizzato un valore medio pari a

$$\left( \frac{\text{volume}}{n \text{ tetraedri}} \right)^{1/3}.$$

La soluzione (5.7) è stata confrontata, in Figura (5.4), con la componente  $x$  della velocità di una simulazione a  $Re_\tau = 15$ , con  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_{1b})$ ,  $\mathbf{f}_x = 1$  e stessa griglia delle simulazioni presentato in precedenza. Le condizioni iniziali sono di fluido fermo,  $\mathbf{u} = 0$  e le condizioni al contorno cicliche in direzione  $x$  e  $y$ .

La figura mostra come la soluzione numerica e quella analitica abbiano lo stesso andamento qualitativo. È possibile affermare che il comportamento numerico del termine di Smagorinsky sia in linea con quello teorico. La piccola differenza di intensità è dovuta alla dissipazione numerica dello schema agli elementi finiti e al fatto che l'ampiezza del filtro  $\bar{\Delta}$  sia stata approssimata con un valore medio, non dipendente dalla coordinata  $z$ .



## Risultati: regime turbolento

I risultati mostrati nel capitolo si riferiscono a simulazioni in regime turbolento. La LES utilizza modelli universali che necessitano di una calibrazione specifica dipendente dalle caratteristiche dell'esperimento.

Lo scopo di queste simulazioni è quello di calibrare i parametri dei modelli di turbolenza introdotti. La difficoltà in questo procedimento è quella di riuscire a trovare la giusta configurazione di parametri che permetta lo svilupparsi della turbolenza. Le Tabelle 6.1 e 6.2 illustrano il processo di calibrazione per cercare la configurazione ideale, per i problemi del flusso nel canale cilindrico e della corrente nel canale piano.

Le grandezze medie analizzate sono medie istantanee nelle direzioni di omogeneità della corrente.

### 6.1 Flusso nel canale cilindrico

Le prime simulazioni effettuate per il flusso nel canale cilindrico utilizzavano condizioni al bordo non cicliche. Alla superficie  $\Sigma_{in}$  veniva applicata una determinata portata, attraverso una condizione di Dirichlet non omogenea, mentre alla superficie  $\Sigma_{out}$  una condizione di sforzo normale nullo. Sulla superficie laterale  $\Sigma_L$ , supposta ferma, veniva imposta la condizione di non scivolamento  $\mathbf{u} = 0$ .

L'imposizione di una portata in ingresso rendeva necessario aumentare le dimensioni del dominio. Per dare lo spazio necessario affinché il profilo corretto si sviluppasse, bisognava triplicare la lunghezza del tubo nella direzione parallela al flusso, con conseguente aumento del costo computazionale.

Il passaggio a condizioni cicliche sulla velocità tra le superfici  $\Sigma_{in}$  e  $\Sigma_{out}$  permette di mantenere la lunghezza del dominio  $L_z = 2\pi$  e quindi diminuire il costo compu-

	C.I.	Dist	$\Delta t$	Modello	$\bar{\Delta}$	$C_S$	Risultato
1	$\mathbf{u}, \mathbf{p} = 0$	$\mathcal{O}(2)$	$5 \cdot 10^{-4}$	no modello	—	—	pb. sviluppo
2	$\mathbf{u}, \mathbf{p} = 0$	$\mathcal{O}(2)$	$5 \cdot 10^{-4}$	Van Driest	$\sqrt[3]{\nu \omega l}$	0.1	pb. sviluppo
3	Stokes	—	$5 \cdot 10^{-4}$	no modello	—	—	forte inst.
4	Stokes	—	$1 \cdot 10^{-4}$	no modello	—	—	forte inst.
5	Stokes	—	$5 \cdot 10^{-4}$	Van Driest	$\sqrt[3]{\nu \omega l}$	0.1	forte inst.
6	Stokes	$\mathcal{O}(5)$	$1 \cdot 10^{-4}$	no modello	—	—	forte inst.
7	Stokes	$\mathcal{O}(5)$	$5 \cdot 10^{-4}$	Van Driest	$\sqrt[3]{\nu \omega l}$	0.1	forte inst.
8	log	5%	$5 \cdot 10^{-4}$	no modello	—	—	forte inst.
9	log	10%	$5 \cdot 10^{-4}$	no modello	—	—	forte inst.
10	log	5%	$5 \cdot 10^{-4}$	Van Driest	$\sqrt[3]{\nu \omega l}$	0.1	inst.
11	log	5%	$5 \cdot 10^{-4}$	Van Driest	diam2	0.1	laminare
12	log	10%	$5 \cdot 10^{-4}$	Van Driest	diam2	0.1	laminare
13	log	$\mathcal{O}(1)$	$5 \cdot 10^{-4}$	Van Driest	$\sqrt[3]{\nu \omega l}$	0.18	crescita dist.

Tabella 6.1: Regime turbolento, flusso nel canale cilindrico,  $Re_\tau = 380$ : parametri e risultati delle simulazioni effettuate con condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, \mathbf{p}) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $\mathbf{f}_z = 4$ .

tazionale. La lunghezza del dominio in direzione  $z$  equivale al valore per cui è stata dimostrata in letteratura l'assenza di correlazione spaziale tra le componenti della velocità.

Velocità e pressione appartengono alla coppia di spazi  $\mathcal{P}_2/\mathcal{P}_1$ . Questa coppia è preferita alla scelta  $\mathcal{P}_{1b}/\mathcal{P}_1$  per garantire una approssimazione più accurata della velocità. La combinazione degli spazi  $\mathcal{P}_2/\mathcal{P}_1$  si dimostra inoltre meno dissipativa. Lo schema di avanzamento temporale scelto è lo schema BDF di secondo ordine. La forzante applicata in direzione  $z$  è  $\mathbf{f}_z = 4$  ricavata da (4.4).

La griglia deve garantire un numero sufficiente di nodi nello strato limite. Il raffinamento avviene in coordinate cilindriche con la funzione  $\tanh(3.5\rho)$ . La griglia è composta da 16650 vertici e 89784 tetraedri. La circonferenza della superficie di ingresso ha 76 nodi ed è ripetuta 30 volte in direzione  $z$ .

I risultati in Tabella 6.1 sono relativi alle simulazioni nel dominio cilindrico con condizioni cicliche per la velocità. La tabella contiene le informazioni relative alla condizione iniziale applicata, colonna *C.I.*, e al relativo disturbo casuale imposto, *Dist*. Il parametro  $\Delta t$  è il passo temporale utilizzato. I dati relativi al tipo di modello utilizzato,  $a$  come viene calcolata l'ampiezza del filtro localmente e al valore della

costante  $C_S$ , sono contenuti rispettivamente nelle colonne *Modello*,  $\bar{\Delta}$ , e  $C_S$ .  $\text{vol}$  è il volume dell'elemento tetraedrico locale, mentre  $\text{diam2}$  il suo diametro calcolato in norma 2. L'ultima colonna della tabella riassume il risultato della simulazione.

Il valore del numero di Reynolds turbolento utilizzato in queste simulazioni,  $Re_\tau = 380$  è equivalente a  $Re \simeq 5600$  basato sulla velocità media (4.7). Il valore corrispondente basato sulla velocità a centro canale è  $Re \simeq 7248$  [2].

I valori calcolati assicurano che il regime delle simulazioni è turbolento.

### 6.1.1 $Re_\tau = 380$ , condizione iniziale casuale

Le simulazioni 1 e 2 della Tabella 6.1 sono caratterizzate da una condizione iniziale di fluido fermo a cui viene aggiunto un disturbo casuale del secondo ordine in tutte le direzioni. Il disturbo viene applicato per permettere alla turbolenza di svilupparsi. Il passo temporale è  $\Delta t = 5e^{-4}$ .

La simulazione 1 è eseguita senza alcun modello di turbolenza, mentre alla simulazione 2 è applicato il modello di Smagorinsky con correzione di Van Driest. L'ampiezza del filtro locale viene calcolata come la radice cubica del volume dell'elemento tetraedrico, mentre il valore di  $C_S = 0.1$ .

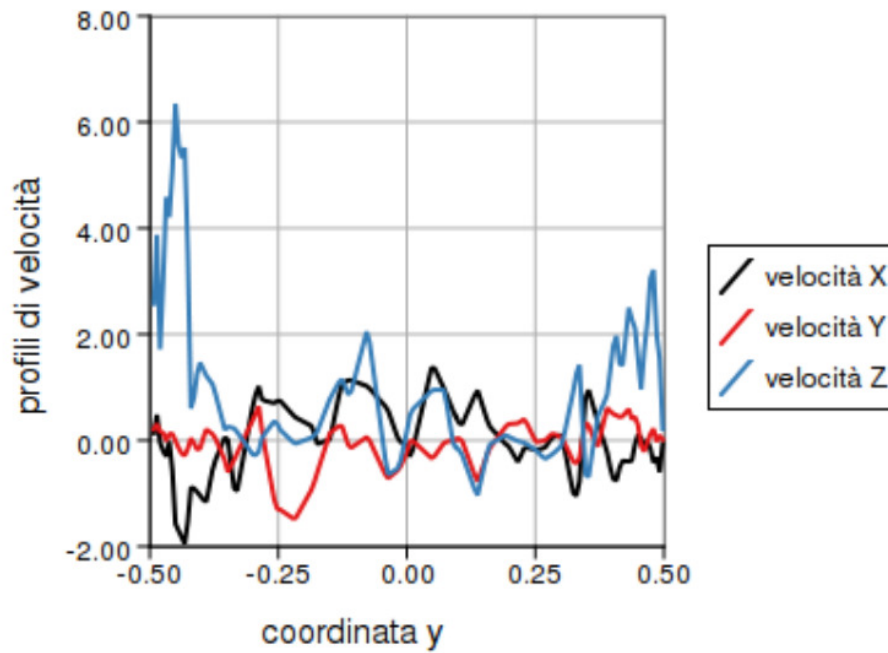
La Figura 6.1 mostra i profili di velocità dopo 200 iterazioni. Il profilo di velocità mostra delle difficoltà a svilupparsi in tempi brevi. La perturbazione random disturba la regolare evoluzione del moto del fluido che necessita di tempi molto lunghi per raggiungere il profilo di regime.

La Figura 6.2 mostra l'evoluzione dell'energia cinetica per le simulazioni 1 e 2. Come si vede in figura l'andamento dell'energia cinetica subisce un brusco cambio di andamento. Questo coincide con la fine della convergenza del metodo iterativo GMRES.

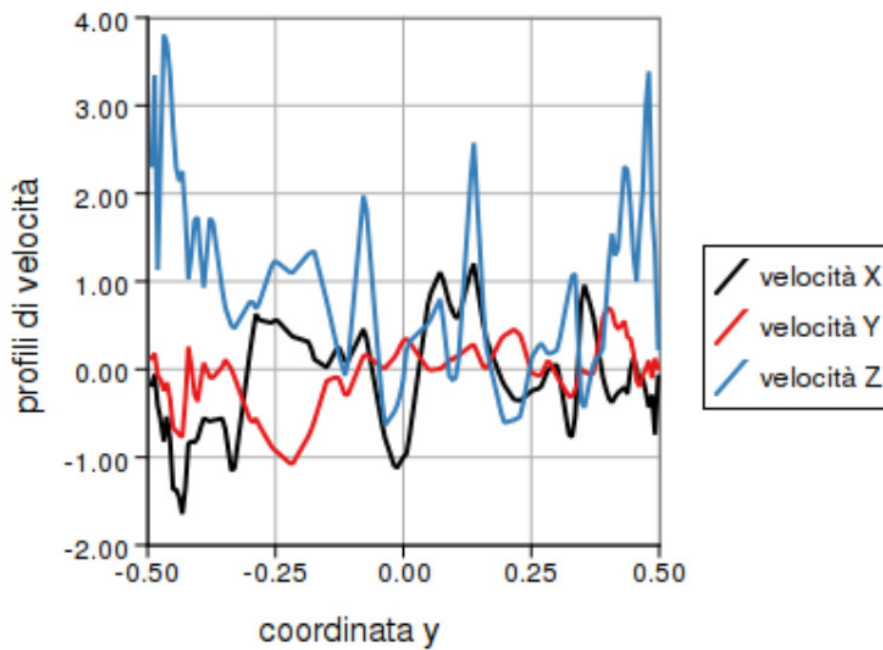
### 6.1.2 $Re_\tau = 380$ , condizione iniziale Stokes

Alle simulazioni 3–7 in Tabella 6.1 viene applicata una condizione iniziale corrispondente alla soluzione del problema di Stokes stazionario associato al problema di Oseen, profilo parabolico in Figura 6.3.

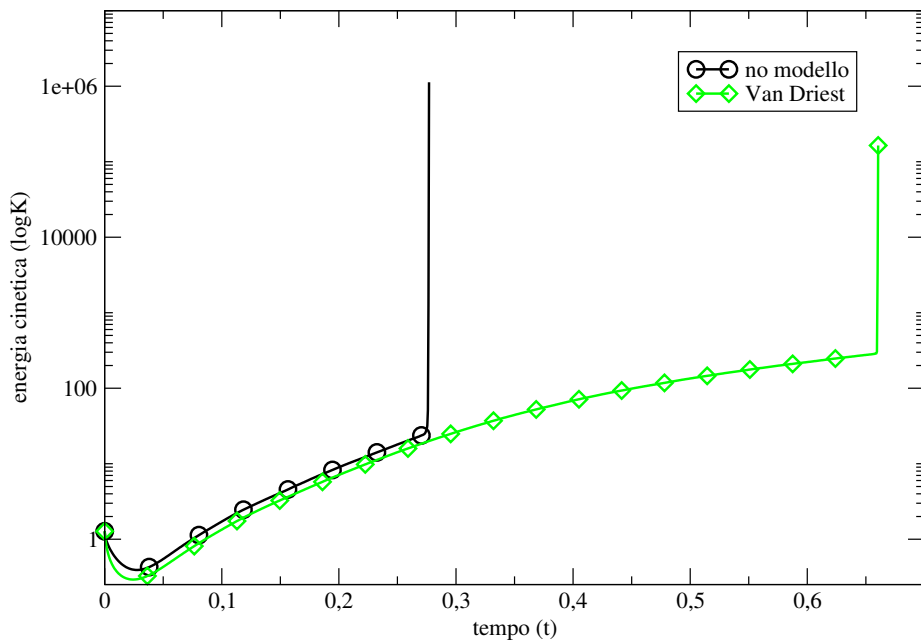
Le simulazioni 3–5 non hanno nessun disturbo casuale, mentre a 6 e 7 viene applicato un disturbo di  $o(5)$ , pari a circa il 5% del valore massimo di velocità. Come mostrato in Tabella 6.1 vengono utilizzati diversi passi temporali. Le simulazioni con stabilizzazione utilizzano il modello di Smagorinsky con correzione di Van Driest. L'ampiezza del filtro locale  $\bar{\Delta} = \sqrt[3]{\text{vol}}$  e  $C_S = 0.1$ .



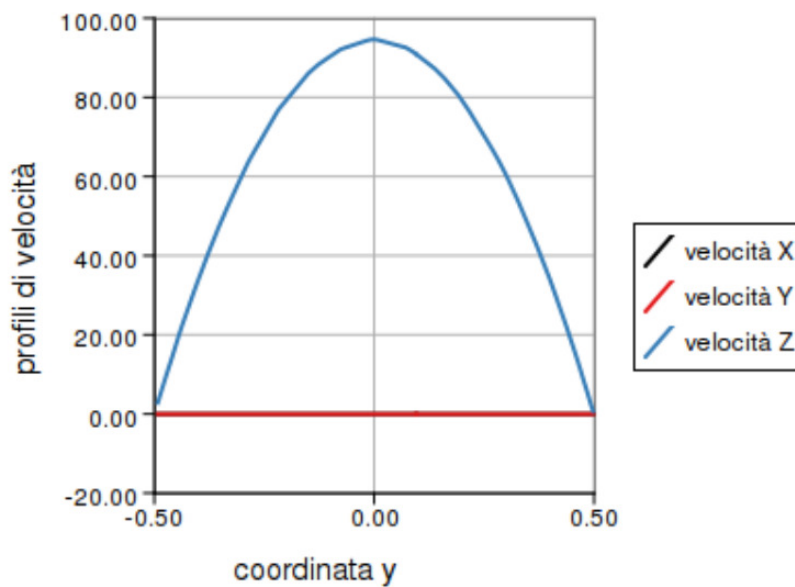
(a) no modello

(b) Van Driest,  $C_S = 0.1$ ,  $\Delta = \text{vol}^{1/3}$ 

**Figura 6.1:** Canale cilindrico  $Re_\tau = 380$ . Profili di velocità istantanei in funzione della coordinata  $y$ , calcolati su un diametro, a partire da condizione iniziale casuale.  $x = 0$ ,  $z = \pi$ ,  $t = 0.1$ . Condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $f_z = 4$ .



**Figura 6.2:** Canale cilindrico  $Re_\tau = 380$ . Energia cinetica in funzione del tempo a partire da condizione iniziale casuale, ordinate in scala logaritmica. Condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $\mathbf{f}_z = 4$ .



**Figura 6.3:** Canale cilindrico  $Re_\tau = 380$ , condizione iniziale Stokes. Soluzione del problema di Stokes stazionario, calcolata su un diametro.

I risultati ottenuti da queste simulazioni mostrano delle forti instabilità. Il metodo risolutivo smette infatti di convergere dopo poche iterazioni. Questo perché la soluzione iniziale applicata impone un flusso con valore massimo pari a circa 5 volte il valore massimo caratteristico del profilo turbolento. Il codice non riesce a smorzare i valori troppo elevati della velocità e la simulazione smette di convergere.

L'aggiunta del disturbo per facilitare lo sviluppo della turbolenza si rivela inefficace. In questo caso la convergenza è mantenuta per un numero ancora più piccolo di iterazioni.

Sarà quindi necessario imporre un profilo iniziale differente.

### 6.1.3 $Re_\tau = 380$ , condizione iniziale logaritmica

Il profilo iniziale applicato alle simulazioni 8 – 13 della Tabella 6.1 corrisponde al profilo logaritmico descritto in [18]:

$$\begin{cases} \mathbf{u}_x = 0, \\ \mathbf{u}_y = 0, \\ \mathbf{u}_z = 2.5 \log \rho^+ + 5.5 & \text{se } \rho^+ > 11.6, \\ \mathbf{u}_z = \rho^+ & \text{se } \rho^+ \leq 11.6, \end{cases} \quad (6.1)$$

dove  $\rho^+$  è la distanza dal bordo in unità di parete (1.8).

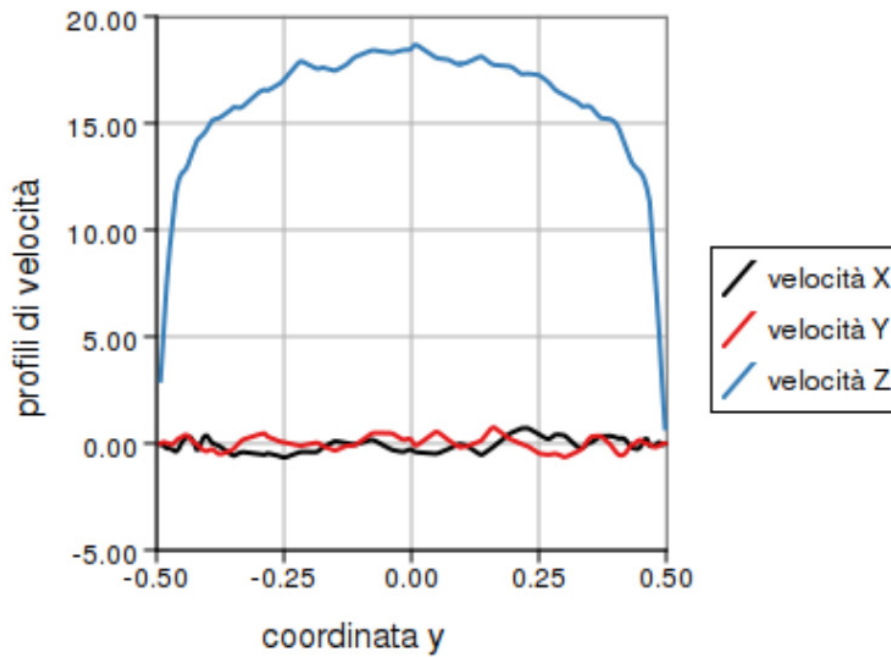
Il profilo (6.1) corrisponde al profilo medio turbolento sviluppato teorico. In questo modo la soluzione iniziale è molto vicina alla soluzione che si desidera ottenere. Alla condizione iniziale è aggiunta una perturbazione, in tutte le direzioni, proporzionale al valore della velocità longitudinale, componente  $z$ . La perturbazione è necessaria affinché si sviluppi la turbolenza. La Figura 6.4 mostra il profilo iniziale (6.1) con disturbo al 5% del valore della velocità longitudinale.

Il passo temporale scelto è  $\Delta t = 5e^{-4}$ .

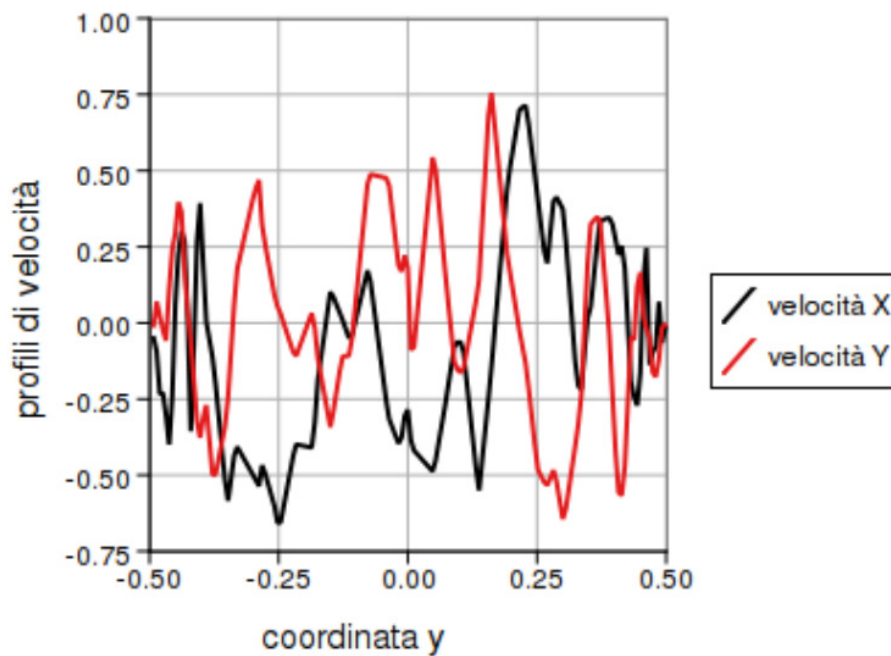
Le simulazioni 8 e 9, senza nessun modello di turbolenza, rispettivamente con un disturbo iniziale del 5% e 10%, si dimostrano instabili e dopo alcune iterazioni non raggiungono più la convergenza. Questo è in linea con i risultati teorici poiché il regime delle simulazioni è turbolento. L'aggiunta del modello di turbolenza si rivela quindi necessaria.

Le simulazioni 10, 11 e 12 utilizzano il modello di Smagorinsky con correzione di Van Driest.

La differenza più importante tra la simulazione 10 e le due successive è relativa all'ampiezza del filtro utilizzato. Mentre la simulazione 10 approssima  $\bar{\Delta} = \sqrt[3]{\nu l}$ ,

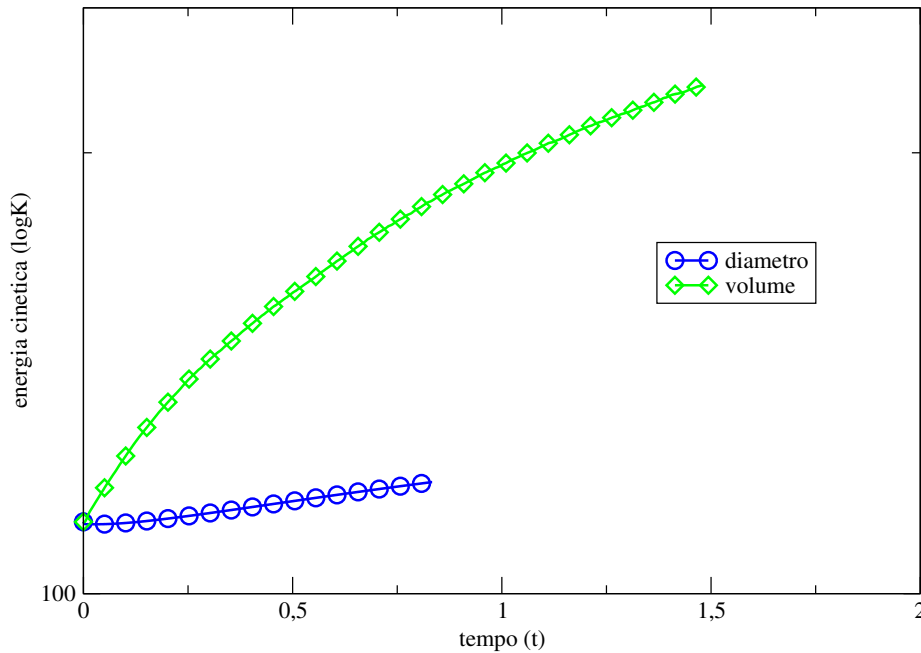


(a) componenti X,Y e Z



(b) dettaglio componenti X e Y

**Figura 6.4:** Canale cilindrico  $Re_\tau = 380$ , condizione iniziale logaritmica con disturbo al 5%.



**Figura 6.5:** Canale cilindrico  $Re_\tau = 380$ . Energia cinetica in funzione del tempo a partire da condizione iniziale logaritmica con disturbo al 5%, ordinate in scala logaritmica. Condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $\mathbf{f}_z = 4$ .

dove  $vol$  è il volume dell'elemento tetraedrico locale, le simulazioni 11 e 12 utilizzano un filtro con ampiezza  $\bar{\Delta} = diam2$ , che corrisponde al diametro calcolato in norma 2 dell'elemento locale.

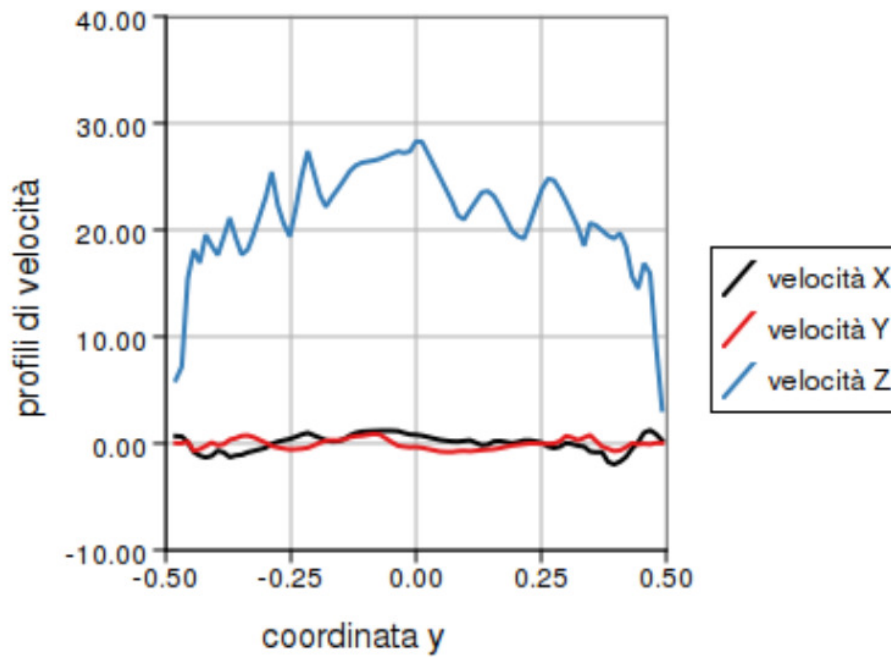
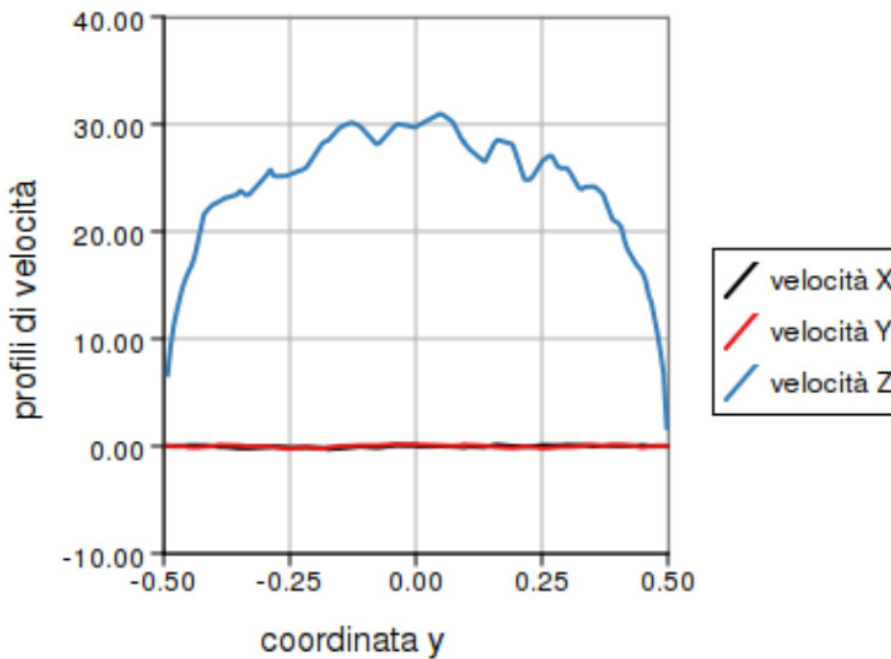
Come mostra la Figura 6.5, a parità di condizioni iniziali, l'energia cinetica della simulazione 10 è superiore alla curva della simulazione 11. Il filtro utilizzato in 10 si rivela quindi meno dissipativo. Il motivo di questo comportamento è legato al tipo di geometria e griglia considerate. I tetraedri deformati per risolvere lo strato limite risultano molto allungati e di conseguenza il valore del diametro è superiore alla radice cubica del volume.

Le soluzioni delle simulazioni 11 e 12 tendono verso un profilo parabolico, sintomo che la soluzione tende a un profilo non turbolento. Inoltre le componenti Y e Z della velocità subiscono uno smorzamento troppo elevato, Figura 6.6b.

I motivi di questo risultato potrebbero essere dovuti all'introduzione di una eccessiva viscosità numerica, utilizzando un valore troppo alto per  $\bar{\Delta}$  o  $C_S$ . Un'altra ragione potrebbe essere legata al tipo di disturbo introdotto. Un disturbo troppo basso nello strato limite potrebbe non essere sufficiente per permettere lo svilupparsi della turbolenza.

Confrontando i profili di velocità nello stesso punto del dominio, allo stesso istante



(a)  $\bar{\Delta} = \text{vol}^{1/3}$ (b)  $\bar{\Delta} = \text{diam}^2$ 

**Figura 6.6:** Canale cilindrico  $Re_\tau = 380$ . Profili di velocità istantanei, calcolati su un diametro, in funzione della coordinata  $y$  per diverse ampiezze del filtro  $\bar{\Delta}$  a partire da condizione iniziale logaritmica.  $x = 0$ ,  $z = \pi$ ,  $t = 1.48$ . Condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $\mathbf{f}_z = 4$ .

temporale e a parità di condizione iniziale, è possibile notare le differenze tra i risultati ottenuti con due diverse ampiezze del filtro, Figura 6.6.

La Figura 6.7 mostra nel dettaglio le componenti istantanee della velocità, per il dominio cilindrico, della simulazione 10 al tempo  $t = 1.48$ .

I valori massimi ottenuti per le componenti  $x$  e  $y$  della velocità e i valori negativi della componente  $z$  risultano troppo elevati. I valori attesi per queste variabili sono infatti circa il 10% del valore massimo raggiunto dalla componente  $z$  della velocità. I valori riscontrati indicano che la viscosità introdotta è troppo bassa.

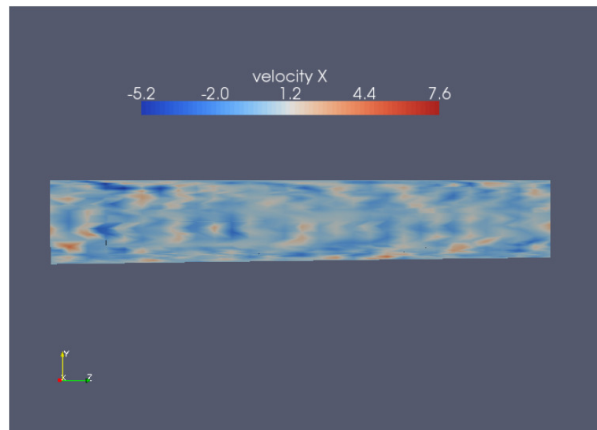
Per questo motivo nella simulazione 13 sarà introdotto un disturbo più elevato nello strato limite e il valore del parametro  $C_S$  sarà scelto in modo da fornire una configurazione intermedia tra le simulazioni 10, 11 e 12.

La simulazione 13 utilizza i parametri  $\bar{\Delta} = \sqrt[3]{\nu \Delta t}$  e  $C_S = 0.18$  per il modello di Smagorinsky con correzione di Van Driest. I profili istantanei si sono rivelati più stabili di quelli ottenuti dalla simulazione 10, ma contemporaneamente meno smorzati di quelli ottenuti da 11 e 12.

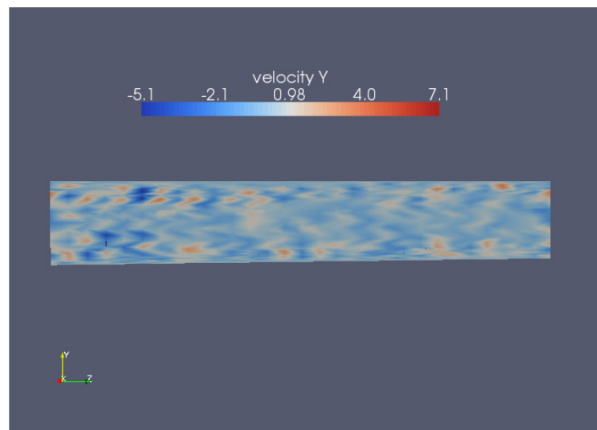
I risultati di 13 sono stati sottoposti a ulteriori analisi. Le componenti trasversali della velocità, in coordinate cartesiane, sono state integrate su piani ortogonali alla direzione  $z$ . Da queste quantità, attraverso relazioni geometriche elementari, sono state ricavate le medie istantanee nello spazio della velocità radiale  $u_r$  e tangenziale  $u_t$ , in funzione della distanza dalla superficie laterale.

La Figura 6.8 mostra l'andamento di  $\langle u_r \rangle$  e  $\langle u_t \rangle$  a differenti istanti temporali. I valori attesi per queste variabili sono prossimi a zero. Questo è un effetto diretto del carattere casuale dei fenomeni turbolenti. I risultati mostrati dalla Figura 6.8 sono di intensità ridotta e quindi in accordo con la teoria.

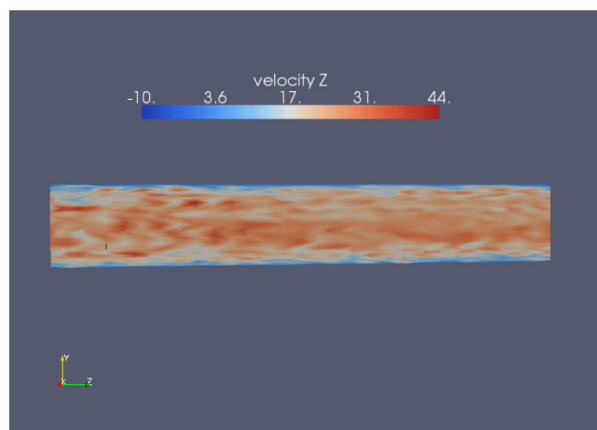
Di maggior interesse è l'analisi delle fluttuazioni istantanee di  $u_r$  e  $u_t$ . Il comportamento evolutivo di queste quantità descritto dalle Figure 6.9a e 6.9b è infatti in linea con i risultati teorici: la fluttuazione al tempo  $t = 0$ , imposta dalla condizione iniziale con disturbo casuale omogeneo, viene smorzata dal modello di turbolenza per poi riprendere intensità, con valori più elevati vicino al bordo.



(a) componente x

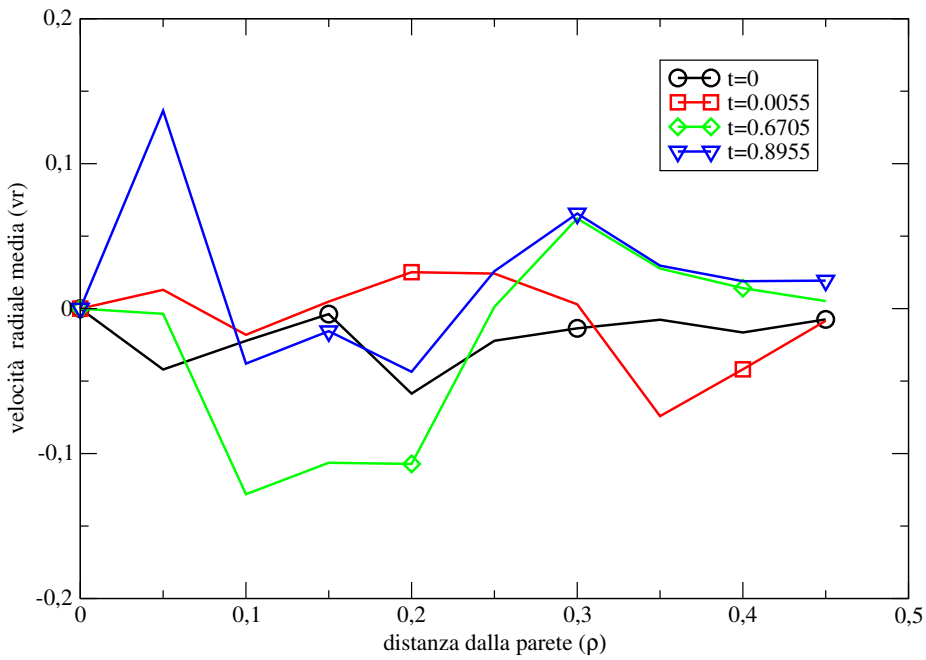


(b) componente y

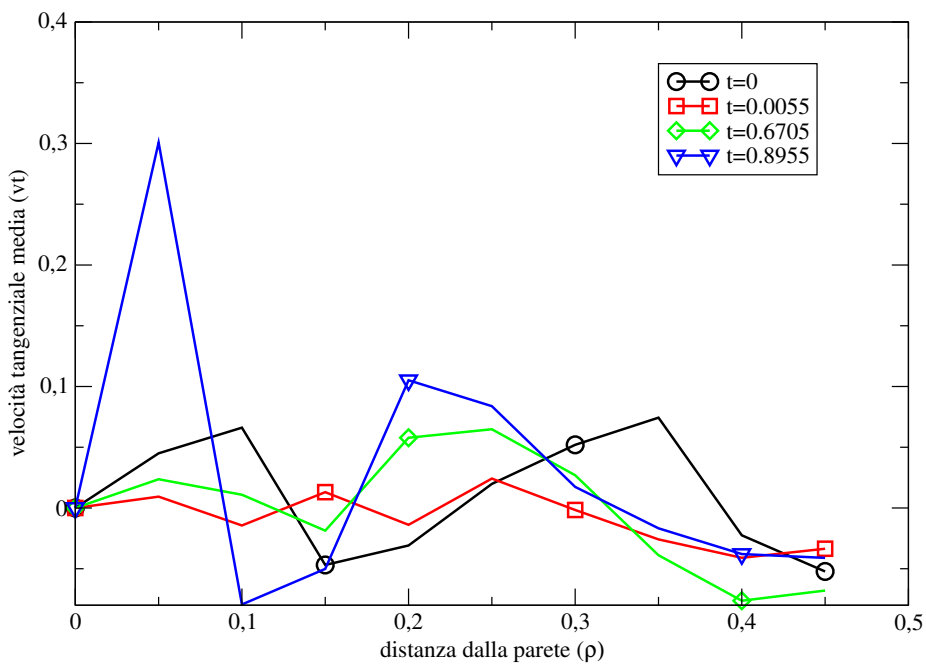


(c) componente z

**Figura 6.7:** Canale cilindrico  $Re_\tau = 380$ . Velocità istantanee sul piano  $yz$  al tempo  $t = 1.48$ ,  $x = 0$ . Condizione iniziale logaritmica con disturbo al 5%. Condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $f_z = 4$ .

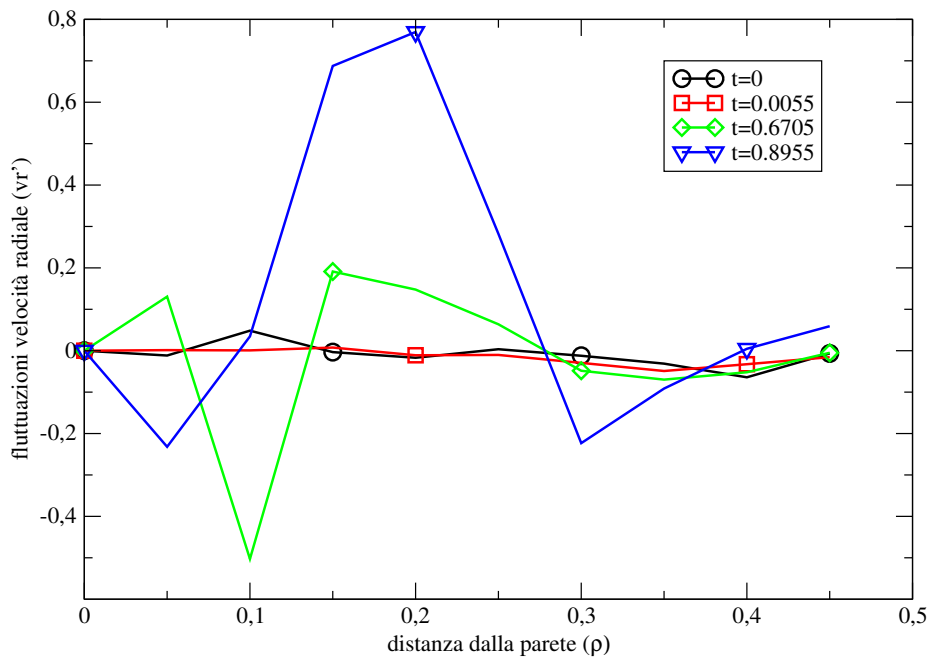


(a) componente radiale

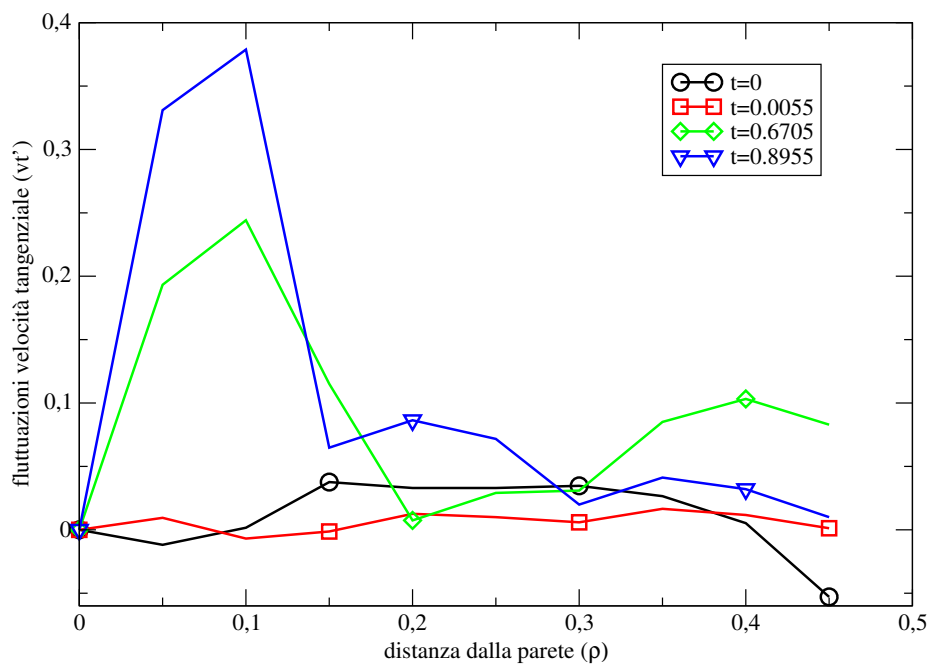


(b) componente tangenziale

**Figura 6.8:** Canale cilindrico  $Re_\tau = 380$ . Velocità medie istantanee in funzione della distanza dal bordo per differenti istanti temporali, a partire da condizione iniziale logaritmica con disturbo  $o(1)$ . Condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $f_z = 4$ .



(a) componente radiale



(b) componente tangenziale

**Figura 6.9:** Canale cilindrico  $Re_\tau = 380$ . Fluttuazioni di velocità istantanee in funzione della distanza dal bordo per differenti istanti temporali, a partire da condizione iniziale logaritmica con disturbo  $o(1)$ . Condizioni al contorno cicliche tra  $\Sigma_{in}$  e  $\Sigma_{out}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $\mathbf{f}_z = 4$ .

	u	Dist	$\Delta t$	Modello	$\bar{\Delta}$	$C_S$	Risultato
1	$\mathcal{P}_2$	5%	$5 \cdot 10^{-4}$	no modello	—	—	forte inst.
2	$\mathcal{P}_2$	10%	$5 \cdot 10^{-4}$	Van Driest	$\sqrt[3]{\text{vol}}$	0.1	forte inst.
3	$\mathcal{P}_2$	5%	$5 \cdot 10^{-4}$	Van Driest	$\sqrt[3]{\text{vol}}$	0.1	inst.
4	$\mathcal{P}_2$	5%	$5 \cdot 10^{-4}$	Van Driest	diam2	0.1	smorz. elevato
5	$\mathcal{P}_2$	o(1)	$5 \cdot 10^{-4}$	Van Driest	diam2	0.1	smorz. elevato
6	$\mathcal{P}_2$	o(2)	$5 \cdot 10^{-4}$	Van Driest	diam2	0.06	inst.
7	$\mathcal{P}_2$	o(2)	$5 \cdot 10^{-4}$	Van Driest	diam2	0.1	qual. accettabile
8	$\mathcal{P}_{1b}$	o(2)	$5 \cdot 10^{-4}$	Van Driest	diam2	0.1	laminare

Tabella 6.2: Regime turbolento, corrente nel canale piano.  $Re_\tau = 180$ : parametri e risultati delle simulazioni effettuate con condizioni al contorno cicliche tra  $\Sigma_{in}$ ,  $\Sigma_{out}$  e  $\Sigma_{1_1}$ ,  $\Sigma_{1_2}$ . Condizione iniziale (4.9). BDF2,  $f_x = 1$ .

## 6.2 Corrente nel canale piano

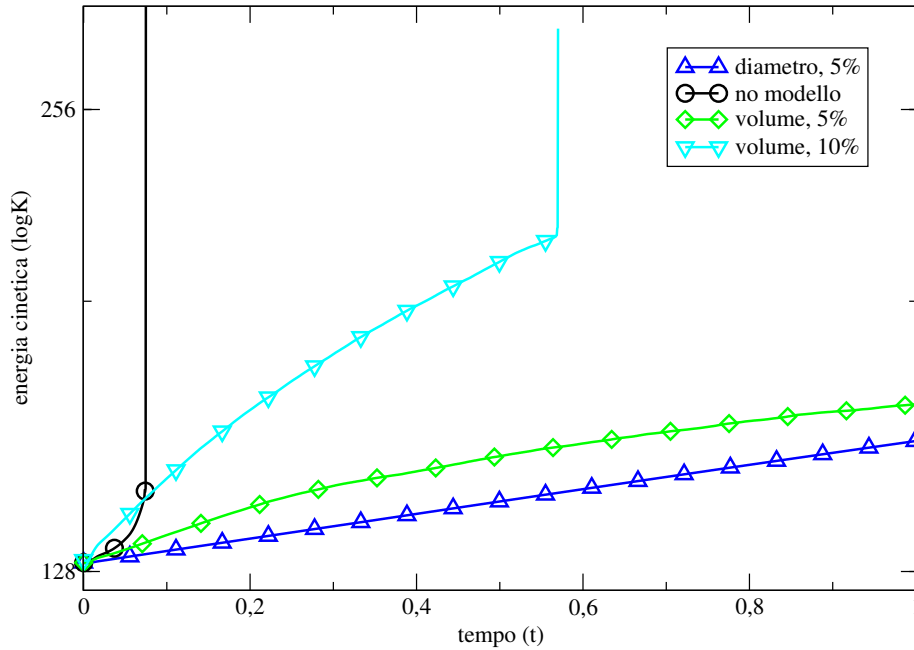
Dopo una analisi dei primi risultati ottenuti nel canale cilindrico, è stato affrontato il problema della corrente in un canale piano e si è deciso di applicare sin dalle prime simulazioni condizioni iniziali di tipo logaritmico.

Le condizioni al contorno per la velocità nel canale piano sono di tipo ciclico tra le superfici  $\Sigma_{in}$ ,  $\Sigma_{out}$  e  $\Sigma_{1_1}$ ,  $\Sigma_{1_2}$ . Lo schema temporale utilizzato è lo schema Backward Difference di ordine 2. Il moto è generato applicando una forzante in direzione  $x$  di valore  $f_x = 1$ .

La pressione appartiene allo spazio a elementi finiti  $\mathcal{P}_1$ , mentre per la velocità sono state utilizzati due differenti spazi a elementi finiti: lo spazio  $\mathcal{P}_2$  e lo spazio  $\mathcal{P}_{1b}$ .

Il numero di Reynolds utilizzato nelle simulazioni è basato sulla velocità d'attrito ed è pari a  $Re_\tau = 180$ . Questo valore corrisponde a un numero di Reynolds basato sulla velocità a centro canale pari a  $Re \simeq 3500$ , superiore al numero di Reynolds critico  $Re_c \simeq 2935$ . La Tabella 6.2, come per il caso del domino cilindrico, mostra le simulazioni principali effettuate al fine di calibrare i parametri del modello. Nella prima colonna della Tabella 6.2 è indicato lo spazio a elementi finiti a cui appartiene la velocità.

La griglia utilizzata è formata da  $26 \times 28 \times 28$  nodi e 109350 tetraedri. Il raffinamento introdotto in direzione  $z$  per garantire un sufficiente numero di nodi nello strato limite è stato fatto mediante la funzione  $\tanh(4z)$ .



**Figura 6.10:** Canale piano  $Re_\tau = 180$ . Energia cinetica in funzione del tempo, a partire da condizione iniziale logaritmica con disturbo percentuale di diverse intensità, ordinate in scala logaritmica. Condizioni al contorno cicliche tra  $\Sigma_{in}$ ,  $\Sigma_{out}$  e  $\Sigma_{l_1}$ ,  $\Sigma_{l_2}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $\mathbf{f}_x = 1$ .

### 6.2.1 $Re_\tau = 180$ , disturbo percentuale

Il gruppo di simulazioni 1–4, in cui  $\mathbf{u} \in \mathcal{P}_2$ , è caratterizzato da una soluzione iniziale logaritmica, corrispondente al profilo medio turbolento sviluppato, a cui è aggiunto un disturbo casuale in ogni direzione, con valore proporzionale al campo di velocità longitudinale.

La simulazione 1 non è stabilizzata da nessun modello di turbolenza. Dopo circa 150 iterazioni, il metodo iterativo che risolve il sistema lineare smette di convergere e la soluzione del problema diventa sempre più instabile fino a esplodere. Si può notare questo andamento divergente nell'ultima parte della curva in nero in Figura 6.10.

Come per il caso cilindrico sono state confrontate a parità di condizioni iniziali le due diverse ampiezze del filtro.

La stabilizzazione aggiunta nelle simulazioni 2–4, tramite l'introduzione del modello di Smagorinsky con correzione di Van Driest, utilizza un parametro  $C_S = 0.1$ .

La simulazione 2, con disturbo iniziale pari al 10% del valore della velocità longitudinale, si dimostra instabile dopo circa 1000 iterazioni. Per questo motivo nelle

simulazioni 3 e 4 la percentuale di disturbo introdotta è stata ridotta al 5%.

La Figura 6.10 mostra che, come per il caso del cilindro, il modello con  $\bar{\Delta} = \text{diam}2$  è più diffusivo di quello che utilizza un valore del parametro  $\bar{\Delta} = \sqrt[3]{\text{vol}}$ . La curva in verde assume infatti valori più elevati della curva in blu, a parità di passo temporale. Il motivo di questo differente andamento è legato alla geometria del problema: i tetraedri utilizzati infatti sono molto allungati in direzione  $z$ . Come conseguenza il valore di  $\bar{\Delta} = \sqrt[3]{\text{vol}}$ , che corrisponde allo spigolo di un cubo equivalente al tetraedro, sarà minore della lunghezza caratteristica del tetraedro calcolata in norma 2,  $\bar{\Delta} = \text{diam}2$ .

L'analisi dei profili di velocità per le simulazioni 3 e 4, ottenuti a parità di condizioni iniziali, mostra delle notevoli differenze, Figura 6.11. Queste differenze sono dovute al tipo di ampiezza del filtro utilizzato, parametro  $\bar{\Delta}$ .

Nella Figura 6.11a, che mostra i profili di velocità della simulazione che utilizza  $\bar{\Delta} = \sqrt[3]{\text{vol}}$ , si nota una eccessiva instabilità dei profili longitudinali, direzione  $z$ , che sono molto frastagliati e presentano zone di alta velocità. I profili delle componenti trasversali della velocità, direzione  $y$  e  $x$ , mostrano un buon andamento.

Analizzando i valori della velocità in tutto il dominio, si notano delle zone in cui sono presenti dei picchi con valori molto elevati. Questo fenomeno si presenta in tutte le direzioni. Per cercare di limitarlo viene eseguita la simulazione 4 che introduce una viscosità numerica maggiore, utilizzando  $\bar{\Delta} = \text{diam}2$ .

I profili della simulazione 4, Figura 6.11b, sono più regolari della precedente, 3, per effetto della maggiore stabilizzazione introdotta. Nonostante i miglioramenti ottenuti in direzione longitudinale la simulazione presenta un eccessivo smorzamento delle componenti trasversali, effetto diretto dell'aumento di viscosità numerica.

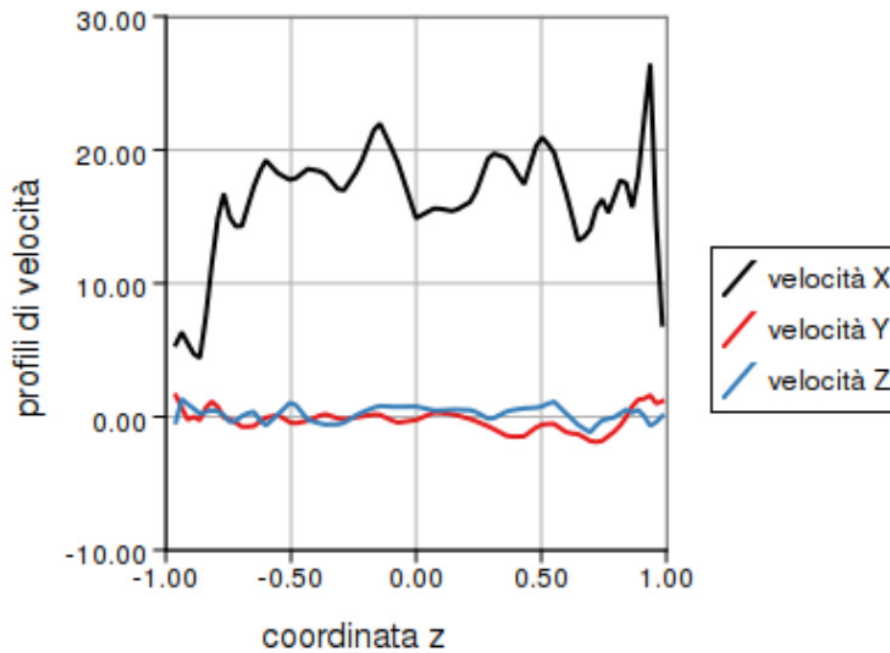
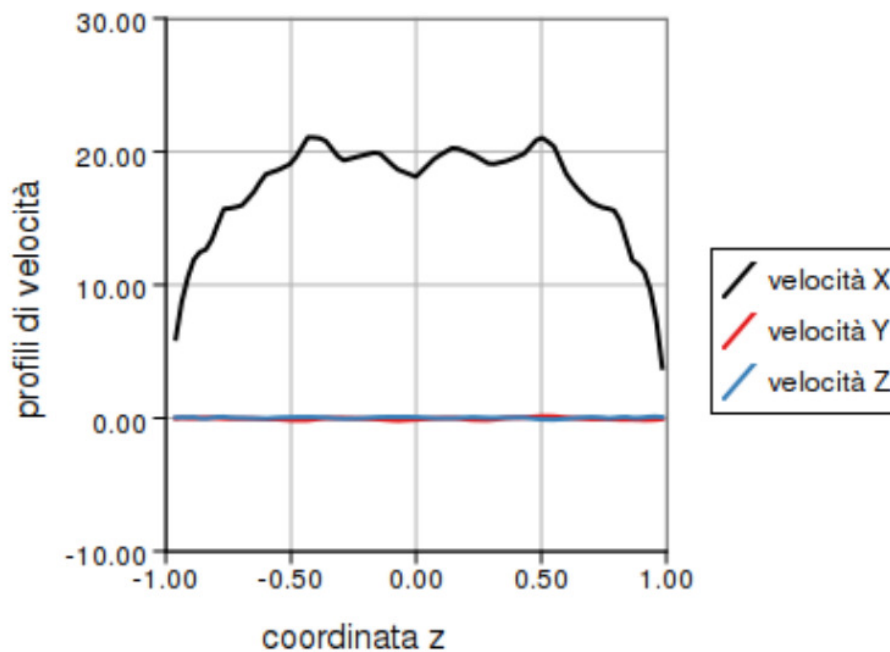
Per le simulazioni successive utilizzeremo la configurazione più stabile,  $\bar{\Delta} = \text{diam}2$ , accoppiata a un aumento del disturbo nello strato limite.

### 6.2.2 $Re_{\tau} = 180$ , disturbo omogeneo

Il disturbo casuale introdotto inizialmente è di fondamentale importanza per permettere alla turbolenza di svilupparsi. Un disturbo troppo basso nello strato limite potrebbe infatti non essere sufficiente a far partire le giuste frequenze di oscillazione e portare allo sviluppo della turbolenza. Il semplice fatto di utilizzare un valore di  $Re_{\tau}$  maggiore del valore critico potrebbe perciò non essere sufficiente.

Per questo motivo si è deciso di applicare alle simulazioni presentate in questa sezione un disturbo omogeneo in tutte le direzioni. In questo modo il disturbo non è più proporzionale al valore della velocità longitudinale, molto bassa in prossimità



(a)  $\bar{\Delta} = \text{vol}^{1/3}$ (b)  $\bar{\Delta} = \text{diam}2$ 

**Figura 6.11:** Canale piano  $Re_\tau = 180$ . Profili di velocità istantanei in funzione della coordinata  $z$  per diverse ampiezze del filtro  $\bar{\Delta}$  a partite da condizione iniziale logaritmica con disturbo percentuale al 5%.  $x = \pi$ ,  $y = 0$ ,  $t = 1$ . Condizioni al contorno cicliche tra  $\Sigma_{in}$ ,  $\Sigma_{out}$  e  $\Sigma_{l_1}$ ,  $\Sigma_{l_2}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $\mathbf{f}_x = 1$ .

dello strato limite.

La simulazione 5 con  $\mathbf{u} \in \mathcal{P}_2$ , utilizza un valore di  $\bar{\Delta} = \text{diam}2$  e  $C_S = 0.1$ . Il modello di turbolenza utilizzato è quello di Smagorinsky con correzione di Van Driest. Il disturbo omogeneo appartiene all'intervallo  $[-1, 1]$ . La Figura 6.12a mostra i profili iniziali a centro canale delle componenti della velocità in direzione  $y$  e  $z$ . La Figura 6.12b mostra l'evoluzione dei profili al tempo  $t = 0.95$ .

Come per la simulazione 4, viene riscontrato un eccessivo smorzamento delle componenti trasversali della velocità. La configurazione dei parametri per questa simulazione si rivela quindi troppo diffusiva. La causa di ciò potrebbe essere attribuita ad un valore troppo elevato della costante  $C_S$  o ad un valore troppo piccolo dell'intensità del disturbo assegnato.

Di conseguenza per la simulazione 6 si è scelto un valore  $C_S = 0.06$  e un disturbo omogeneo appartenente a  $[-2, 2]$ . Con questa nuova configurazione i risultati ottenuti sono però instabili con picchi di velocità elevati e profili istantanei molto frastagliati in direzione longitudinale.

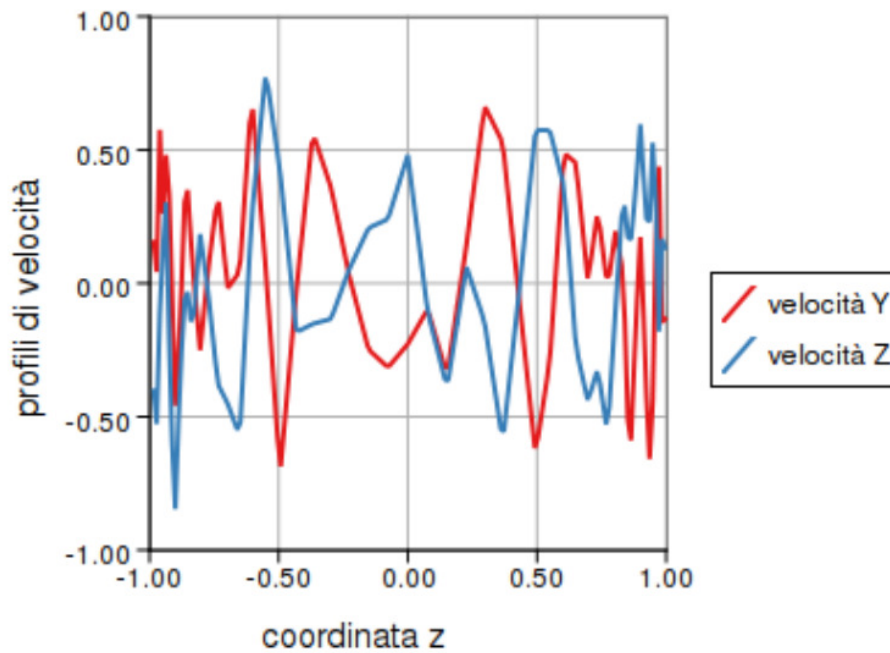
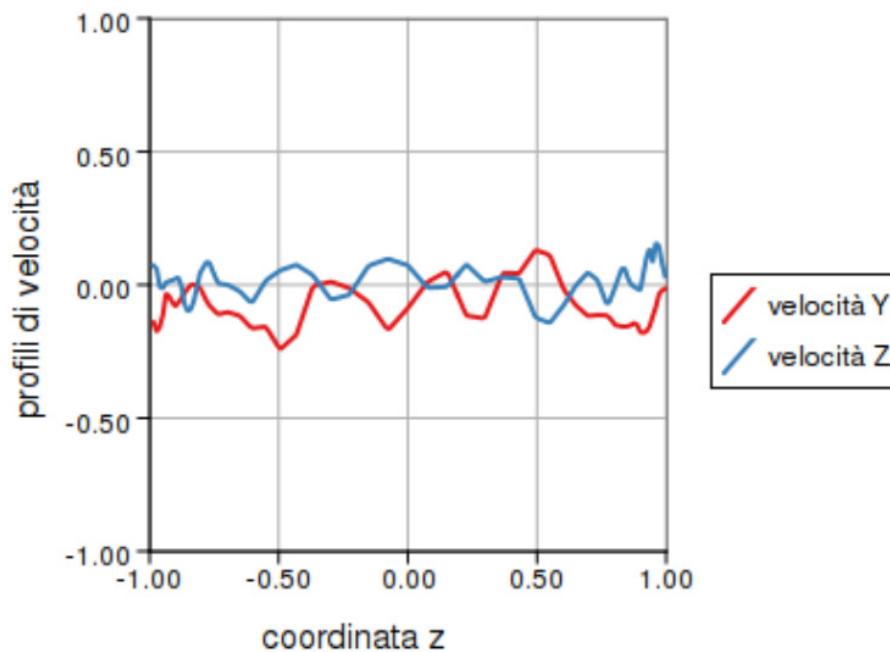
La simulazione 7 utilizza gli stessi parametri della precedente ad eccezione del valore di  $C_S = 0.1$  per cercare di fornire una maggior stabilizzazione.

I risultati ottenuti verranno confrontati con quelli della simulazione 8 che utilizza la stessa configurazione della 7, con velocità in  $\mathcal{P}_{1b}$ .

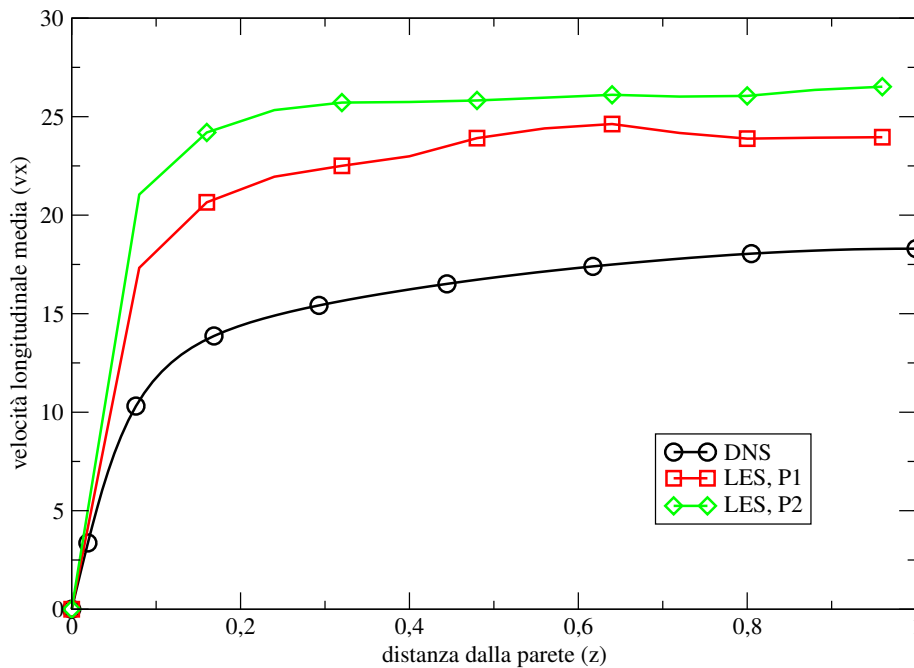
Per le due simulazioni verranno effettuate le medie spaziali su piani  $xy$  per poter confrontare i risultati ottenuti con quelli mostrati nel database [15], che riporta i risultati della simulazione numerica diretta per problemi nel canale piano a differenti valori di  $Re_\tau$ .

La Figura 6.13 mostra i profili di velocità mediati su piani  $xy$  per le simulazioni 7 e 8 e il profilo medio a regime estratto dal database [15] per lo stesso valore di  $Re_\tau$ . I profili di velocità medi delle simulazioni ottenute con LifeV seguono l'andamento qualitativo della curva in nero, corrispondente al profilo medio a regime, ma assumono valori superiori. Il motivo di questo potrebbe essere dovuto al passo temporale a cui sono ricavate le curve in rosso e verde. Le simulazioni 7 e 8 non hanno ancora raggiunto il regime e quindi i valori medi riscontrati, superiori ai valori a regime, sono ammissibili. La Figura 6.13 evidenzia anche la maggior capacità diffusiva dello spazio  $\mathcal{P}_{1b}$  rispetto allo spazio  $\mathcal{P}_2$ , poiché la curva in rosso assume valori minori di quella in verde.

La Figura 6.14 analizza le componenti oscillatorie della velocità in direzione  $y$  e  $z$ , calcolate come la varianza della velocità nelle due direzioni, e propone un confronto con i valori dello sforzo di Reynolds,  $R_{ij} = \overline{u'_i u'_j}$  di [15].

(a)  $t = 0$ (b)  $t = 0.95$ 

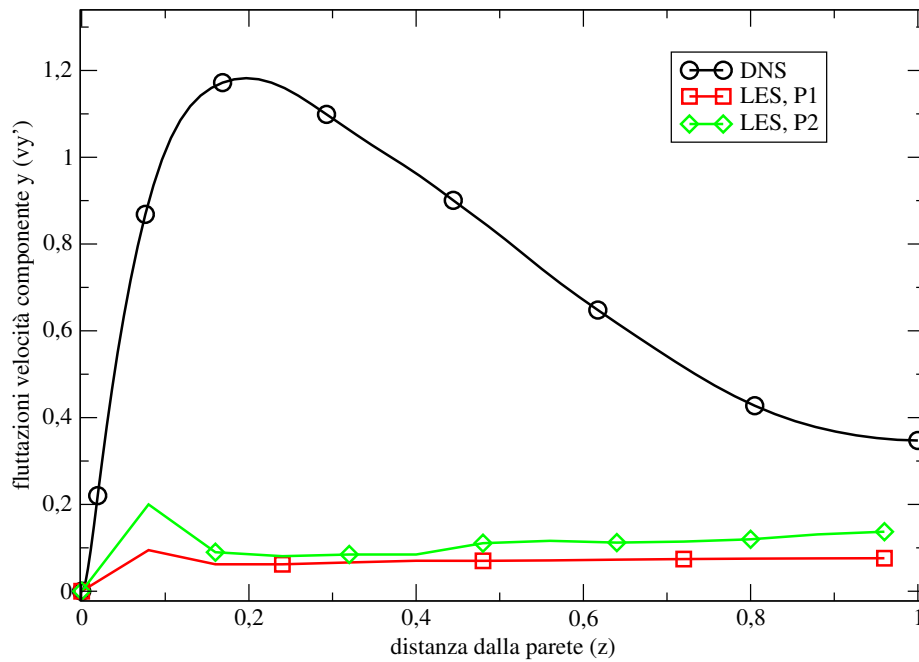
**Figura 6.12:** Canale piano  $Re_\tau = 180$ . Profili di velocità istantanei in funzione della coordinata  $z$ , a partire da condizione iniziale logaritmica con disturbo omogeneo  $\mathcal{O}(1)$ .  $x = \pi, y = 0$ . Condizioni al contorno cicliche tra  $\Sigma_{in}, \Sigma_{out}$  e  $\Sigma_{l_1}, \Sigma_{l_2}$ .  $(\mathbf{u}, p) \in (\mathcal{P}_2, \mathcal{P}_1)$ , BDF2,  $f_x = 1$ .



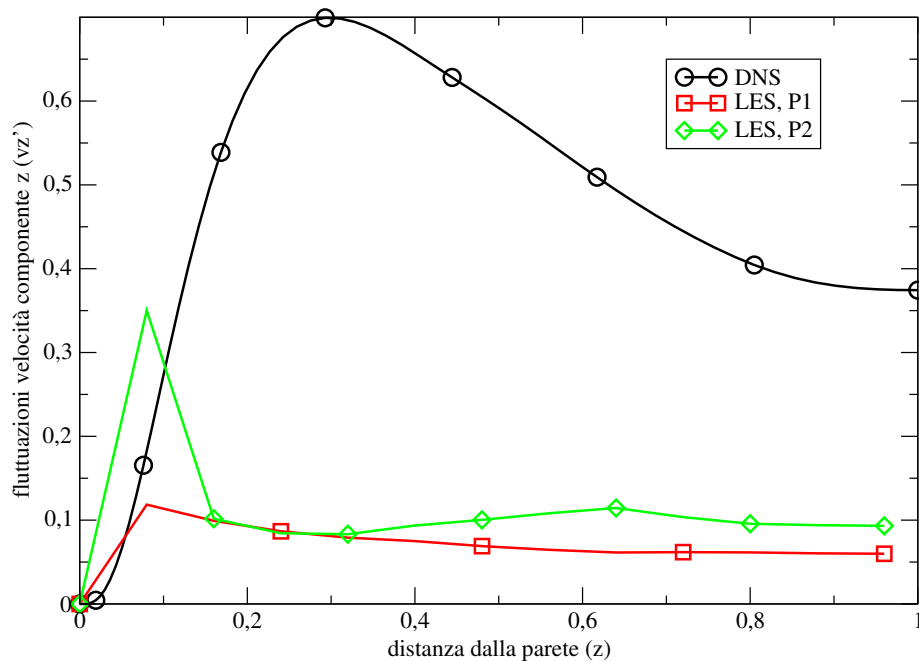
**Figura 6.13:** Canale piano  $Re_\tau = 180$ . Profili di velocità istantanei medi in funzione della distanza dal bordo  $z$  a partire da condizione iniziale logaritmica con disturbo omogeneo  $o(2)$ ,  $t = 1$ . Condizioni al contorno cicliche tra  $\Sigma_{in}$ ,  $\Sigma_{out}$  e  $\Sigma_{l_1}$ ,  $\Sigma_{l_2}$ . BDF2,  $f_x = 1$ .

Analizzando la Figura 6.14 si nota come l'andamento delle oscillazioni, più elevate vicino a parete e più deboli al centro, sia in accordo con il risultato mostrato dalla curva in nero. Il confronto effettuato ha tuttavia solamente un valore qualitativo perché i dati delle curve in rosso e verde sono relativi a simulazioni non ancora a regime. La quantità mostrata nella curva in nero inoltre non corrisponde esattamente al dato ricavato per le simulazioni 7 e 8. Il comportamento atteso dalle tre curve deve tuttavia essere lo stesso.

Per entrambe le simulazioni si registra un aumento dei valori massimi di velocità. La componente  $x$  della velocità per la simulazione 8 tende al profilo parabolico tipico del regime laminare per effetto della maggior diffusività dello spazio  $\mathcal{P}_{1b}$ . La simulazione 7 mantiene invece un profilo istantaneo qualitativo, in direzione  $x$ , accettabile.



(a) direzione y



(b) direzione z

**Figura 6.14:** Canale piano  $Re_\tau = 180$ . Fluttuazioni di velocità istantanee in funzione della distanza dal bordo  $z$  a partire da condizione iniziale logaritmica con disturbo omogeneo  $o(2)$ ,  $t = 1$ . Condizioni al contorno cicliche tra  $\Sigma_{in}$ ,  $\Sigma_{out}$  e  $\Sigma_{l1}$ ,  $\Sigma_{l2}$ . BDF2,  $f_x = 1$ .

## Conclusioni e sviluppi futuri

In questa tesi Magistrale sono state introdotte le funzioni necessarie per la simulazione a grandi vortici, LES, nella libreria per la simulazione numerica agli elementi finiti LifeV, scritta in linguaggio C++.

Il filtraggio introdotto dalla tecnica LES permette un minor costo computazionale rispetto alla simulazione numerica diretta e soluzioni più accurate di quelle ottenute con la tecnica RANS.

Il modello di turbolenza implementato è il modello di Smagorinsky. Il limite principale di questo modello è legato alle sue proprietà sovradiffusive. Il rimedio al problema si ottiene aggiungendo il termine correttivo di Van Driest al termine di Smagorinsky. Le funzioni responsabili della costruzione del termine correttivo sono state introdotte nella libreria.

La libreria contiene un test per le simulazioni in dominio cilindrico che è stato utilizzato come punto di partenza per il problema del flusso nel cilindro. Un caso test su domini poligonali è stato implementato per permettere la trattazione del problema della corrente nel canale piano.

Il modello di turbolenza è stato applicato prima a simulazioni in regime laminare e in seguito a problemi in regime turbolento. I parametri delle simulazioni sono stati scelti da risultati presenti in letteratura.

In regime laminare il modello ha confermato la sua natura diffusiva. Alcune simulazioni hanno mostrato delle instabilità in disaccordo con la configurazione corrispondente ai parametri utilizzati. I motivi di queste instabilità sembrano essere legati allo spazio agli elementi finiti utilizzato e al tipo di condizione iniziale assegnata.

In questo lavoro il modello di turbolenza è stato impiegato su griglie non strutturate. Poiché queste procedure sono tuttora oggetto di studi, le simulazioni presentate in

regime turbolento sono utilizzate per la calibrazione del modello. Perché la turbolenza riesca a svilupparsi è necessaria la giusta combinazione di parametri fisici, condizioni al contorno e parametri relativi al modello.

Nel corso del lavoro sono stati fatti notevoli passi avanti nella procedura di calibrazione del modello e nella qualità dei risultati ottenuti, nonostante il regime di turbolenza pienamente sviluppata non sia stato raggiunto.

Guardando al futuro gli spunti offerti da questo lavoro sono molteplici.

Una ulteriore analisi dei risultati in regime laminare sarebbe necessaria per confermare definitivamente le ipotesi avanzate sulla natura delle instabilità.

In regime turbolento il proseguimento naturale del lavoro svolto è la continuazione del processo di calibrazione del modello di turbolenza.

Dal punto di vista dell'implementazione di nuovo codice vi sono due direzioni percorribili. L'introduzione del modello di Germano [5] nella libreria, che permetterebbe soluzioni più accurate. La seconda direzione è legata alle prestazioni del codice. È infatti necessario un lavoro di programmazione volto a migliorare i tempi di simulazione.

## Bibliografia

- [1] A. Abbà, *Simulazione numerica di correnti turbolente col metodo della Large Eddy Simulation*, Ph.D. thesis, Ingegneria Aerospaziale, Politecnico di Milano, 1991.
- [2] A. AGARD, *Selection of Test Cases for the Validation of Large-Eddy Simulations of Turbulent Flows*, Tech. report, Technical Report AGARD-AR-345, NATO, 1998.
- [3] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag New York, Inc., 1991.
- [4] E. Burman, M.A. Fernández, and P. Hansbo, *Continuous interior penalty finite element method for Oseen's equations*, *SIAM journal on numerical analysis* **44** (2007), no. 3, 1248–1274.
- [5] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot, *A dynamic subgrid-scale eddy viscosity model*, *Physics of Fluids A: Fluid Dynamics* **3** (1991), no. 7, 1760–1765.
- [6] C. Geuzaine and J.F. Remacle, *Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities*, *International Journal for Numerical Methods in Engineering* **79** (2009), no. 11, 1309–1331.
- [7] F. Hecht, *Freefem++ manual*, (2011).
- [8] M.A. Heroux, R.A. Bartlett, V.E Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, R.B. Lehoucq, K.R. Long, R.P. Pawlowski, E.T. Phipps, A.G. Salinger, H.K. Thornquist, R.S. Tuminaro, J.M. Willenbring, A. Williams, and K.S. Stanley,



- An overview of the Trilinos project*, ACM Trans. Math. Softw. **31** (2005), no. 3, 397–423.
- [9] V. John, *Large Eddy simulation of turbulent incompressible flows: analytical and numerical results for a class of les models*, vol. 34, Springer Verlag, 2004.
- [10] G. Karypis, K. Schloegel, and V. Kumar, *ParMETIS: Parallel Graph Partitioning and Sparse Matrix Ordering Library Version 3.2*, University of Minnesota, Minneapolis (2011).
- [11] W. Kollmann, R.C. McCallen, and J.M. Leone Jr, *An examination of les filtering within the finite element method*, Communications in numerical methods in engineering **18** (2002), no. 7, 513–528.
- [12] A.N. Kolmogorov, *Dissipation of energy in the locally isotropic turbulence*, Comptes rendus (Doklady) de l'Académie des Sciences de l'U.R.S.S. **32** (1941), no. 1890, 16–18.
- [13] CP Kothandaraman and R. Rudramoorthy, *Fluid mechanics and machinery*, New Age International, 2007.
- [14] D.K. Lilly, *A proposed modification of the Germano subgrid-scale closure method*, Physics of Fluids A: Fluid Dynamics **4** (1992), no. 3, 633–634.
- [15] R.D. Moser, J. Kim, and N.N. Mansour, *Direct numerical simulation of turbulent channel flow up to  $Re_\tau = 590$* , Physics of Fluids **11** (1999), 943–945.
- [16] A. Quarteroni, *Numerical models for differential problems*, Springer Verlag, 2008.
- [17] P. Sagaut, *Large Eddy simulation for incompressible flows: an introduction*, Springer Verlag, 2006.
- [18] H. Schlichting, *Boundary layer theory*, Pergamon Press, London, 1955.
- [19] L. Valdetaro, *Dispense del corso di Teoria, Modellistica e Simulazione della Turbolenza*, Allievi di Ingegneria Matematica - Laurea Specialistica - Indirizzo Scienze Computazionali per l'Ingegneria, A.A. 2007/08.