

**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA AEROSPAZIALE

**SIMULAZIONI NUMERICHE DI FLUSSI
TURBOLENTI NON STAZIONARI IN GEOMETRIE
COMPLESSE**

Relatore

Prof. Francesco Picano

Laureando

Marco Giardina Papa

Correlatore

Ing. Federico Dalla Barba

ANNO ACCADEMICO 2023-2024

Data di laurea 11/07/2024

Sommario

La modellazione di flussi turbolenti in geometrie complesse rappresenta una sfida importante nell'ambito della fluidodinamica computazionale. Una delle problematiche principali riguarda la rappresentazione della geometria all'interno del dominio computazionale. L'obiettivo principale del presente lavoro di tesi è lo sviluppo di un metodo numerico che implementi un *Signed Distance Field* per la rappresentazione della geometria nel dominio di calcolo, in combinazione con una tecnica *Immersed Boundary Method* per simulare l'interazione della geometria con il flusso, e la simulazione dei flussi turbolenti con una *Large Eddy Simulation* modellata a parete. Alla base del codice sviluppato vi è il solutore *CaNS* (*Canonical Navier-Stokes*), un metodo numerico per simulazioni rapide e parallelizzabili di flussi turbolenti. Il metodo numerico viene validato attraverso due casi di riferimento: il flusso su cilindro, caso bidimensionale, e il flusso su sfera, estensione tridimensionale del primo. Il confronto è effettuato attraverso il calcolo del coefficiente di drag al variare del numero di Reynolds e la dimensione della zona di ricircolo generata dalla geometria. Il metodo viene quindi applicato a diverse geometrie complesse per illustrarne le potenzialità.

Abstract

The modeling of turbulent flows in complex geometries represents a significant challenge in computational fluid dynamics. One of the main issues concerns the representation of the geometry within the computational domain. The main objective of this thesis is the development of a numerical method that implements a Signed Distance Field for the representation of geometry in the computational domain, combined with an Immersed Boundary Method to simulate the interaction of the geometry with the flow, and the simulation of turbulent flows using a wall-modeled Large Eddy Simulation. The numerical method is based on the CaNS (Canonical Navier-Stokes) solver, a numerical method for fast and parallelizable simulations of turbulent flows. The numerical method is validated through two reference cases: the flow over a cylinder, a two-dimensional case, and the flow over a sphere, a three-dimensional extension of the former. The comparison is carried out by calculating the drag coefficient as a function of the Reynolds number and the size of the recirculation zone generated by the geometry. The method is then applied to several complex geometries to illustrate its potential.

Indice

1	Introduzione	1
2	Immersed Boundary Method	5
2.1	Origini del Metodo	7
2.2	Categorie dei metodi IBM	11
2.2.1	Continuous Forcing Approach	12
2.2.2	Discrete Forcing Approach	17
3	Signed Distance Field	23
3.1	Formato STL	23
3.2	SDF: Definizione e Proprietà	25
3.3	Modello DeepSDF e modulo <i>mesh-to-sdf</i>	27
3.3.1	Modulo <i>mesh-to-sdf</i>	28
4	Flussi Turbolenti e Modello LES	29
4.1	Turbolenza	29
4.1.1	Quantità Medie	30
4.1.2	Cascata di Energia	31
4.1.3	Teoria di Kolmogorov	33
4.2	Modellistica della Turbolenza	35
4.2.1	Modello LES	36
4.3	Wall-Modeled LES	39
5	Metodo Numerico	41
5.1	Solutore <i>CaNS</i>	42

5.1.1	Implementazione e Message-Passing Interface (MPI)	43
5.2	Immersed Boundary Method	44
5.3	Signed Distance Field	49
6	Validazione Codice	51
6.1	Flusso su Cilindro	52
6.1.1	Calcolo SDF	52
6.1.2	Simulazione e Risultati	54
6.2	Flusso su Sfera	59
6.2.1	Calcolo SDF	59
6.2.2	Simulazione e Risultati	61
7	Risultati Qualitativi	65
7.1	Flusso su Automobile	66
7.2	Flusso in Ambiente Urbano	70
8	Conclusioni	73

Elenco delle figure

2.1	Confronto metodo <i>body-fitted</i> e IBM	6
2.2	Rappresentazione del confine solido nella griglia computazionale	8
2.3	Configurazioni di interfaccia solido-liquido	8
2.4	Intersezione tra griglia euleriana e <i>immersed boundary</i>	9
2.5	Cella MAC	9
2.6	Classificazione della cella euleriana rispetto all' <i>immersed boundary</i>	10
2.7	Flusso incompressibile in dominio 2D con <i>immersed boundary</i>	13
2.8	Risultati della simulazione di Peskin	15
2.9	Esempi di <i>distribution functions</i>	15
2.10	Approccio <i>ghost-cell</i>	18
2.11	Approccio <i>cut-cell</i>	20
3.1	<i>Facet</i> del formato STL	24
3.2	Tassellatura STL	24
3.3	Esempi di <i>Signed Distance Field</i>	26
3.4	Proprietà del <i>Signed Distance Field</i>	26
3.5	Esempio modello DeepSDF	27
3.6	Uso di <i>mesh-to-sdf</i>	28
4.1	Quantità medie	31
4.2	Valore medio e quantità fluttuante	32
4.3	Cascata di energia	35
4.4	<i>Large Eddy Simulation</i> : velocità filtrata	37
4.5	Schema WMLES.	40

5.1	Schema del metodo numerico	41
5.2	Decomposizione MPI	43
5.3	Griglia Euleriana e Lagrangiana	45
5.4	Punti Lagrangiani adiacenti	46
5.5	Retrazione dei punti della griglia	46
5.6	SDF al variare della risoluzione	49
6.1	Dominio computazionale.	51
6.2	Geometria STL del cilindro.	52
6.3	SDF del cilindro.	53
6.4	Parametri della zona di ricircolo.	54
6.5	Flusso su cilindro, risultati $Re = 20$	55
6.6	Flusso su cilindro, risultati $Re = 40$	56
6.7	Flusso su cilindro, risultati $Re = 200$	57
6.8	Geometria STL della sfera.	59
6.9	<i>Contour Plot</i> per quattro valori di SDF della sfera.	60
6.10	C_D vs Re per flusso su sfera.	61
6.11	Grafici <i>streamlines</i> flusso su sfera	62
6.12	C_D vs <i>timestep</i>	63
7.1	SDF automobile	66
7.2	Grafici velocità automobile	67
7.3	Grafici <i>streamlines</i> automobile	68
7.4	Dettaglio <i>streamlines</i> automobile	69
7.5	SDF ambiente urbano	70
7.6	Grafici velocità ambiente urbano	71
7.7	Componenti velocità ambiente urbano	72

Elenco delle tabelle

3.1	Lista di proprietà del SDF.	26
6.1	Confronto con letteratura della dimensione della zona di ricircolo e coefficiente di Drag per $Re = 20$ e $Re = 40$	58
6.2	Confronto con letteratura del coefficiente di Drag per $Re = 200$	58
6.3	Confronto con letteratura della dimensione della zona di ricircolo e coefficiente di Drag per $Re = 100$	63

Capitolo 1

Introduzione

I flussi turbolenti non stazionari in geometrie complesse rappresentano una delle sfide più affascinanti e complesse nell'ambito della fluidodinamica computazionale. Questi flussi sono caratterizzati da una dinamica intricata e imprevedibile, che richiede approcci avanzati e sofisticati per essere modellata e compresa. La capacità di simulare accuratamente tali flussi è fondamentale per una vasta gamma di applicazioni ingegneristiche, che spaziano dall'aerodinamica all'ingegneria civile, passando per le applicazioni biomediche e ambientali.

In questo contesto, la tecnica *Immersed Boundary Method* (IBM) si è affermata come una strategia potente e flessibile per gestire geometrie complesse senza la necessità di conformare la griglia di calcolo ai contorni del corpo solido. Questa metodologia consente di simulare flussi intorno a oggetti di forma irregolare in modo più efficiente rispetto ai metodi tradizionali basati su griglie strutturate. Inoltre, il comportamento del flusso turbolento viene qui simulato attraverso una LES (*Large Eddy Simulation*) modellata a parete (WMLES): questo modello riproduce accuratamente il comportamento dei flussi turbolenti in prossimità delle pareti e funziona in modo ottimale in combinazione con la tecnica IBM.

Una delle problematiche principali nelle simulazioni CFD su geometrie complesse riguarda il passaggio dalla geometria generata da un CAD alla sua rappresentazione nel dominio computazionale: lo sviluppo di metodi per diminuire il tempo computazionale necessario per eseguire questo passaggio risulta essere ancora argomento di ricerca da studi scientifici. Nel presente lavoro di tesi l'input della geometria nel sistema di simu-

lazione viene gestito attraverso un *Signed Distance Field* (SDF), campo che fornisce una rappresentazione continua delle distanze dai contorni delle superfici solide. Il SDF è particolarmente utile per il metodo IB, in quanto facilita l'implementazione delle condizioni al contorno e la gestione delle interfacce tra fluido e solido. Recentemente sono stati pubblicati vari studi che implementano la tecnica IBM accoppiata con il SDF, come ad esempio Zhang [33], che ha implementato il codice *sdf-ibm* basato sul software *OpenFOAM*¹ oppure Lai *et al* [14], i quali hanno implementato un modello con SDF per la simulazione di flussi multifase. Il SDF è un campo scalare che viene calcolato in tutto il dominio computazionale, fornendo informazioni sulla distanza dalla geometria per ogni punto. Le sue applicazioni sono molteplici:

1. Utilizzando il gradiente di SDF vengono calcolate le normali alla geometria e ciò consente di ottenere una nuvola di punti e con rispettive normali che vengono utilizzate dal metodo IBM.
2. Il SDF può essere utilizzato insieme alla tecnica IBM per il calcolo della frazione di volume di solido in una cella del dominio.
3. Può essere calcolata, tramite SDF, la distanza dalla parete per un flusso in regime turbolento.
4. Viene anche utilizzato per costruire volume di controllo e stencil di discretizzazione vicino alla geometria.

Il SDF viene generato a partire da una geometria in formato STL e può essere scelta la risoluzione della griglia di calcolo in questo primo passaggio: ciò consente un filtraggio della geometria in base alla disponibilità di punti della griglia, ottenendo, allo stesso modo, una rappresentazione diretta della geometria all'interno del dominio di calcolo.

Il presente lavoro di tesi si propone di implementare il SDF nel solutore *CaNS*, metodo numerico per simulazione turbolenti, in combinazione con una tecnica IBM e il modello WMLES: questo consente di ottenere un modello in grado di simulare flussi turbolenti a grandi numeri di Reynolds e implementare la presenza di geometrie complesse all'interno del dominio di calcolo.

¹Software CDF open source sviluppato da OpenCFD dal 2004

La tesi è divisa in due parti. La prima comprende i primi tre capitoli e fornisce prima una descrizione teorica delle varie tecniche IBM (Capitolo 2); poi vengono descritte le proprietà di SDF e le modalità con le quali viene calcolato a partire da una geometria in formato STL (Capitolo 3); segue una trattazione della modellistica dei flussi in regime turbolento, prestando maggiore attenzione verso il modello LES utilizzato nel codice (Capitolo 4).

La seconda parte è dedicata al metodo numerico. Il Capitolo 5 descrive in dettaglio il funzionamento del codice sviluppato. Nei Capitoli 6 e 7, il metodo numerico viene validato attraverso due casi numerici e, infine, vengono presentate alcune simulazioni qualitative su geometrie complesse per testare le potenzialità del metodo sviluppato.

Capitolo 2

Immersed Boundary Method

Con il termine *Immersed Boundary Method* (IBM) si fa riferimento ad una categoria di tecniche numeriche utilizzate per la simulazione di flussi viscosi su corpi le cui frontiere non sono conformi alle rispettive griglie computazionali [16]. Al fine di evidenziare le differenze che esistono tra un approccio convenzionale e una tecnica IBM si può considerare l'esempio riportato in Figura 2.1:

- (A) Il metodo convenzionale, detto *body fitted*, prevede la generazione di una griglia che copre la frontiera Γ_b ; questa viene poi utilizzata come condizione al contorno per creare una seconda griglia all'interno del volume Ω_f occupato dal fluido. Successivamente si può utilizzare un metodo alle differenze finite o ai volumi finiti¹ per applicare le equazioni di bilancio e risolvere il problema.

- (B) La tecnica IBM può prevedere la generazione di una griglia sull'*immersed boundary* (IB) del corpo ma questa, al contrario del metodo precedente, non ha alcuna influenza sulla creazione della griglia Cartesiana all'interno del volume del corpo. Poiché la griglia della superficie non risulta conforme al corpo, incorporare le condizioni al contorno richiede una modifica delle equazioni nelle vicinanze dell'IB e le varie tecniche IBM risolvono questo problema in modo diverso. Infine, come nel caso precedente, si utilizzano dei metodi numerici per la discretizzazione delle equazioni di bilancio e la risoluzione del problema.

¹Due tecniche numeriche utilizzate per la risoluzione di equazioni differenziali, sia ordinarie (ODE) sia alle derivate parziali (PDE).

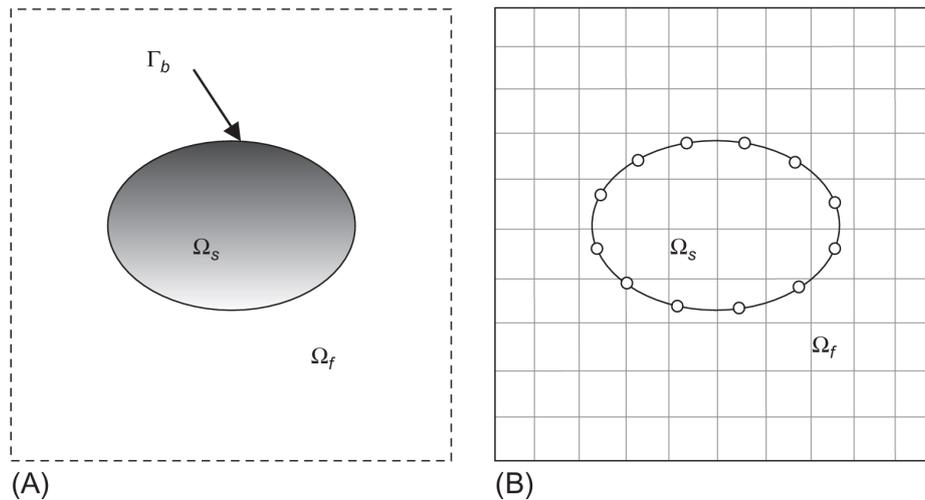


Figura 2.1: (A) Flusso su un corpo generico. Il corpo occupa il volume Ω_s con frontiera Γ_b . Il volume del fluido è definito da Ω_f . (B) Schema del corpo immerso in un griglia cartesiana sulla quale viene eseguita la simulazione.

Da questa breve descrizione risulta evidente che l'imposizione delle condizioni al contorno per le tecniche IBM non è semplice e, di conseguenza, si possono avere dei problemi di accuratezza quando viene applicato il simulatore numerico; con il metodo *body-fitted* si ha un allineamento tra la griglia Cartesiana e la griglia sulla superficie del corpo e questo consente un controllo migliore sulla risoluzione della griglia, portando a dei vantaggi per flussi con alti numeri di Reynolds.

Ciononostante, la generazione di una griglia che sia conforme al corpo è solitamente molto oneroso in termini di calcolo numerico: l'obiettivo è costruire una griglia che fornisca una risoluzione locale adeguata con il numero minimo di punti della griglia totale. Per geometrie semplici questo requisito è facile da rispettare ma aumentando la complessità si può avere un deterioramento della qualità della griglia, il che può influenzare negativamente la precisione e le proprietà di convergenza del solutore. Proprio qui sta il grande vantaggio dell'approccio IBM: per una simulazione eseguita su una griglia cartesiana non conforme al corpo, la complessità e la qualità della griglia non sono significativamente influenzate dalla complessità della geometria.

Questo metodo è stato introdotto per la prima volta dal matematico americano Charles Samuel Peskin negli anni settanta per simulare il flusso di sangue all'interno del cuore [22]. Nonostante la versatilità del metodo implementato da Peskin, la sua adozione nella

comunità più ampia della fluidodinamica computazionale (CFD) è stata lenta. Inizialmente fu utilizzato per problemi nel campo della meccanica cardiovascolare, principalmente da Peskin, dai suoi collaboratori e dai suoi studenti fino alla fine degli anni '80; ciò significa che, per molto tempo, questo potente metodo è rimasto nascosto e solo negli anni '90 iniziò a prendere piede anche fuori dal campo biomedico.

Questo capitolo è diviso in due sezioni: Nella prima si andrà a indagare sulle origini di questo metodo, andando a considerare degli articoli scientifici storici che hanno contribuito allo sviluppo del metodo IBM; nella seconda si andranno a definire le due categorie principali di IBM, ovvero i metodi *continuous forcing approach* e *discrete forcing approach*. La descrizione del metodo adottato nelle simulazioni svolte in questa tesi verrà presentata nel paragrafo 5.2.

2.1 Origini del Metodo

Per rintracciare le radici della tecnica IBM si deve ritornare alla fine degli anni cinquanta quando i computer digitali iniziarono ad essere adottati da vari laboratori nazionali negli Stati Uniti e vennero svolte le prime simulazioni su una vasta gamma di problemi di flusso e trasferimento di calore [19].

Inizialmente le prime simulazioni di fluidodinamica furono condotte su una griglia cartesiana che non poteva considerare geometrie curve. Si prenda in considerazione, ad esempio, la simulazione della scia di von Kármán generata da una piastra piana condotta negli anni 60 [6]: la griglia computazionale in Figura 2.2 è composta da celle di dimensioni δx e δy che vengono utilizzate per il calcolo delle quantità di pressione e velocità tramite il metodo delle differenze finite. Si può notare che le parti della mesh che rappresentano il solido, raffigurate con una linea più spessa, devono necessariamente essere conformi alla griglia e ciò rappresenta un limite al metodo.

Un primo tentativo per superare questo limite fu condotto da Rich e Blackman nel 1962 i quali utilizzarono un metodo Euleriano alle differenze finite (chiamato *fluid-in-cell*) per la simulazione di flussi 2D comprimibili [24]. Nello studio fecero una distinzione tra due configurazioni dell'interfaccia solido-liquido e proposero diverse metodologie per la simulazione: La prima è quella riportata in Figura 2.3a, che chiamarono *barriers at cell*

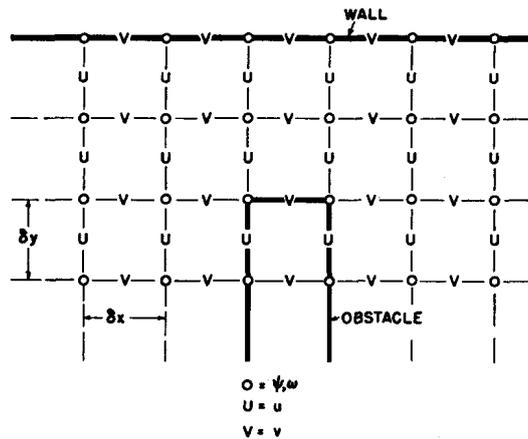


Figura 2.2: Porzione della griglia computazionale che mostra la relazione tra il *solid boundary* (linea spessa) e la mesh (linea chiara) [6].

boundaries, e la seconda è quella di Figura 2.3b, detta *arbitrary linear barriers within a cell*. Per il primo caso definirono delle celle fittizie all'interno del corpo per imporre le condizioni al contorno del muro, e per la cella all'angolo utilizzarono due griglie fittizie a seconda se l'angolo fosse affrontato orizzontalmente o verticalmente. Per il caso generale della Figura 2.3b sono state identificate 12 configurazioni e, per ciascuna di esse, è stato calcolato un insieme di cinque grandezze geometriche, inclusa l'area del fluido e la frazione fluido/solido su ciascun lato, per pesare i termini forzanti delle equazioni e calcolare i flussi corretti attraverso le facce della cella.

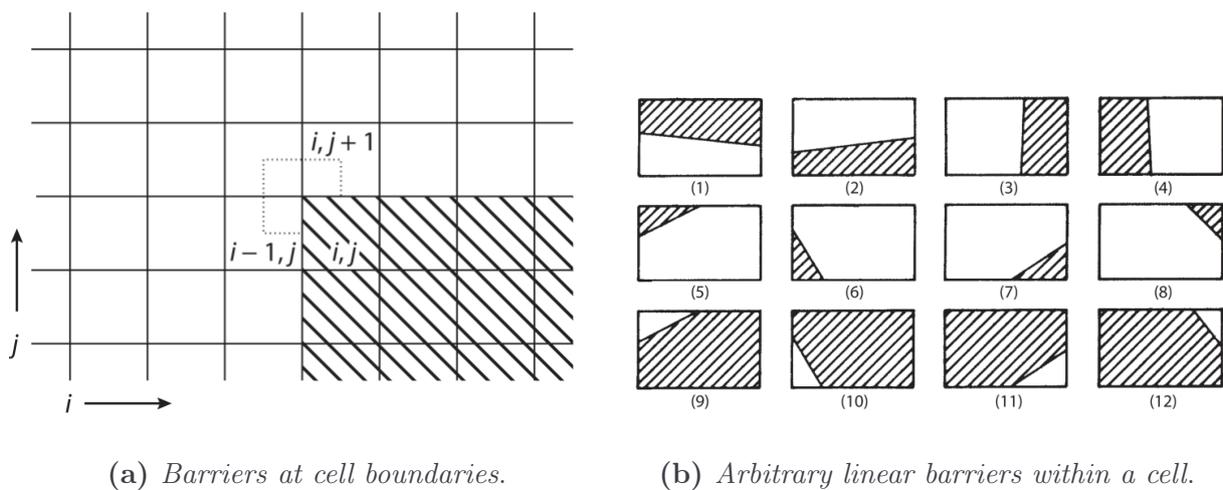


Figura 2.3: Le due configurazioni dell'interfaccia solido-liquido presentate da Rich e Blackman [24].

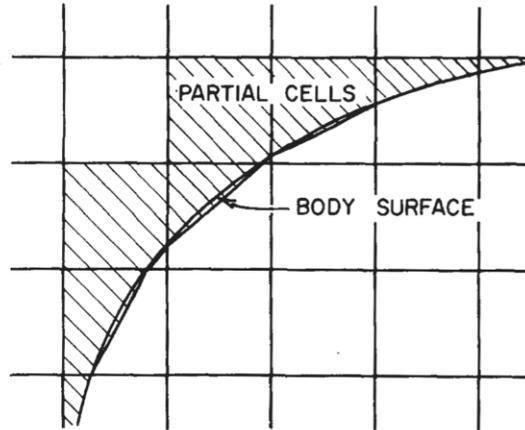


Figura 2.4: Intersezione tra griglia euleriana e *immersed boundary* [7].

Nello studio del 1966 condotto da Gentry *et al.* [7] ci fu un miglioramento del metodo di Rich e Blackman: basandosi sempre su un metodo Euleriano alle differenze finite venne introdotta una maggiore flessibilità nell'uso delle *solid boundaries* riconoscendo che l'utilizzo parziale di una cella della griglia computazionale può causare dei problemi di stabilità se la dimensione di questa è molto piccola. La tecnica che implementarono nei casi presenti dello studio fu quella di ridurre il Δt locale per la simulazione nelle celle parziali (Figura 2.4) e questo risultò essere sufficiente per l'ottenimento di soluzioni stabili.

Successivamente Vieceili presentò nel 1969 il primo metodo per trattare le pareti con curvature arbitrarie [30]. Alla base dello studio vi è il metodo *Marker-And-Cell* (MAC) presentato da Harlow e Welsh nel 1965 [31]: in questo modello le particelle vengono utilizzate come *marker* che individuano il materiale in una mesh che a sua volta definisce la posizione della superficie libera del fluido. La griglia MAC è composta da celle: Su ognuna di queste viene calcolata la pressione p al centro e le tre componenti di velocità u_x , u_y e u_z sulle facce come è possibile vedere in Figura 2.5.

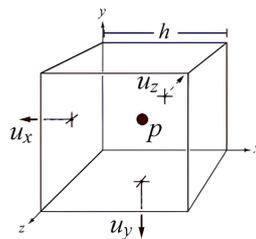


Figura 2.5: Cella MAC.

Il metodo di Viecelli tratta l'IB come delle superfici libere ² e impone la condizione di non-penetrazione attraverso di essi introducendo un termine di forzatura proporzionale al flusso: Successivamente questo metodo fu implementato anche per *moving boundary* utilizzando sempre il metodo MAC [29].

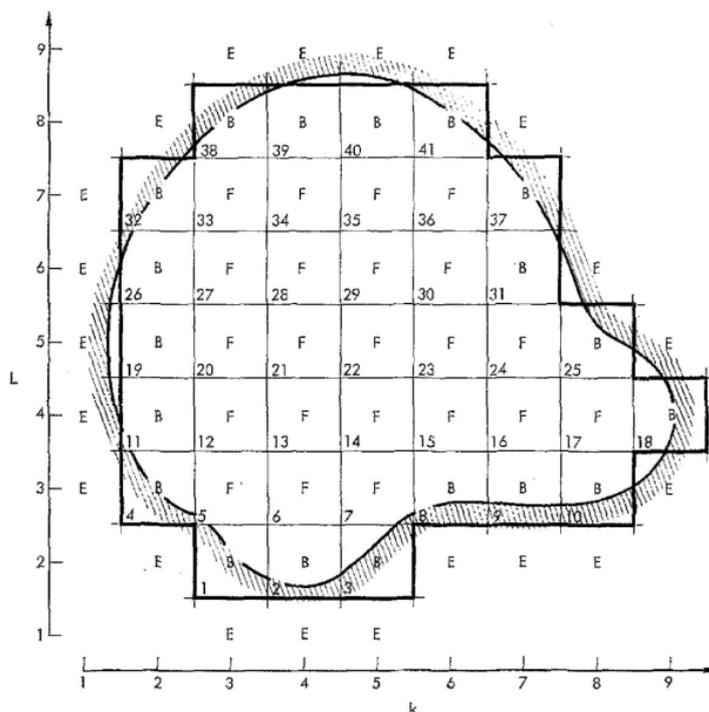


Figura 2.6: Classificazione della cella euleriana rispetto all'*immersed boundary* [29].

Confrontando il contenuto di questi vecchi articoli con gli attuali IBM, si deve riconoscere che tutti gli elementi fondamentali erano già stati sviluppati entro la fine degli anni '60. Il problema dell'epoca era la limitata potenza di calcolo dei computer.

²Una superficie libera (*free surface*) separa un fluido da un altro mezzo, solitamente aria, dove la pressione del fluido è uguale alla pressione atmosferica

2.2 Categorie dei metodi IBM

L'imposizione delle condizioni al contorno sull'IB è il fattore chiave nello sviluppo di una tecnica IBM. Questo aspetto non solo definisce l'identità di una tecnica IBM, ma anche la sua distinzione rispetto alle altre. Riprendendo l'esempio illustrato in Figura 2.1 consideriamo una simulazione di flusso incompressibile governata dalle seguenti equazioni:

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) + \nabla p - \mu \nabla^2 \vec{u} = 0 \quad , \quad (2.1)$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } \Omega_f \text{ e} \quad (2.2)$$

$$\vec{u} = \vec{u}_\Gamma \quad \text{in } \Gamma_b \quad (2.3)$$

ove $\vec{u}(\vec{x}, t)$ è la velocità del fluido, $p(\vec{x}, t)$ è la pressione, i coefficienti ρ e μ sono la densità e la viscosità dinamica del fluido. Il solido occupa lo spazio Ω_b con contorno Γ_b e il dominio fluido è dato da Ω_f . Il sistema può anche essere scritto nel seguente modo:

$$\mathcal{L}(\vec{U}) = 0 \quad \text{in } \Omega_f \quad (2.4)$$

$$\vec{U} = \vec{U}_\Gamma \quad \text{in } \Gamma_b \quad (2.5)$$

ove $\vec{U} = (\vec{u}, p)$ e \mathcal{L} è un operatore che rappresenta le equazioni di Navier-Stokes, ovvero l'Equazione 2.1.

I metodi convenzionali, come il *body fitted*, prevedono una discretizzazione dell'Equazione 2.4 in una griglia conforme al corpo ove le condizioni al contorno, rappresentate dall'Equazione 2.5, sono imposte direttamente. Al contrario le tecniche IBM prevedono una discretizzazione dell'equazione 2.1 su una griglia cartesiana non conforme al contorno e le condizioni al contorno vengono imposte indirettamente tramite una modifica dell'equazione 2.3, la quale rappresenta la condizione di *no-slip*. In generale questa modifica si traduce in un'introduzione di una forza nelle equazioni di bilancio al fine di riprodurre l'effetto delle condizioni al contorno: questa operazione porta alla divisione delle tecniche IBM in due gruppi, ovvero *continuous forcing approach* e *discrete forcing approach*.

Utilizzando il primo approccio, la *forcing function* \vec{f} viene introdotta nell'equazione della quantità di moto nel seguente modo:

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) + \nabla p - \mu \nabla^2 \vec{u} = \vec{f} \quad , \quad (2.6)$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } \Omega_f \text{ e} \quad (2.7)$$

$$\vec{u} = \vec{u}_\Gamma \quad \text{in } \Gamma_b \quad (2.8)$$

e queste vengono successivamente applicate all'intero dominio ($\Omega_f + \Omega_b$). Esistono molti metodi per determinare la forzante esterna \vec{f} , successivamente le equazioni vengono discretizzate su una griglia cartesiana, portando al seguente sistema di equazioni:

$$[L]\{\vec{U}\} = \{\vec{f}\} \quad (2.9)$$

e questo viene risolto sull'intero dominio.

Nel secondo approccio le equazioni 2.1-2.3 vengono prima discretizzate su una griglia cartesiana senza tenere conto dell'IB. Successivamente, la discretizzazione nelle celle vicine all'IB viene regolata per tenere conto della sua presenza, introducendo la *forcing function* \vec{f}' . La differenza rispetto al primo approccio sta nel fatto che la forzante esterna viene introdotta dopo la discretizzazione delle equazioni. Il sistema si scrive nel seguente modo:

$$[L]\{\vec{U}\} = \{\vec{f}'\} \quad (2.10)$$

ove il termine forzante \vec{f}' tiene conto della procedura effettuata.

Una caratteristica interessante del *continuous forcing approach* è che viene formulato indipendentemente dalla discretizzazione spaziale sottostante. D'altra parte, il *discrete forcing approach* dipende molto dal metodo di discretizzazione. Tuttavia, ciò consente un controllo diretto sull'accuratezza numerica, sulla stabilità e sulle proprietà di conservazione discrete del solutore.

2.2.1 Continuous Forcing Approach

Tra i metodi che fanno uso del *continuous forcing approach* bisogna fare una distinzione tra *elastic boundaries* e *rigid boundaries* poiché richiedono dei trattamenti differenti.

Elastic Boundaries

In questa categoria rientrano tutte le tecniche che discendono dal metodo IBM originale introdotto da Peskin.

Lo studio condotto da Peskin si focalizzava sull'interazione fluido struttura ³ all'interno delle valvole cardiache. Il metodo che utilizzò per studiare il problema era una modifica di quello implementato da Viccelli [29], quest'ultimo fu ritenuto inadatto poiché richiedeva che il moto del flusso fosse prescritto. Di conseguenza Peskin elaborò un metodo che differiva completamente dagli altri approcci contemporanei.

Innanzitutto Peskin introdusse una forza di volume nell'equazione della quantità di moto per trattare gli sforzi indotti dal fluido sull'*immersed boundary*, non considerò le condizioni al contorno come un vincolo:

$$\frac{\partial \rho \vec{u}}{\partial t} + \vec{u} \cdot \nabla \rho \vec{u} = -\nabla p + \mu \nabla^2 \vec{u} + \vec{f}_b \quad (2.11)$$

ove ρ e μ sono la densità e la viscosità dinamica del fluido, \vec{u} e p sono velocità e pressione e \vec{f}_b è la forza di volume usata per applicare la condizione al contorno.

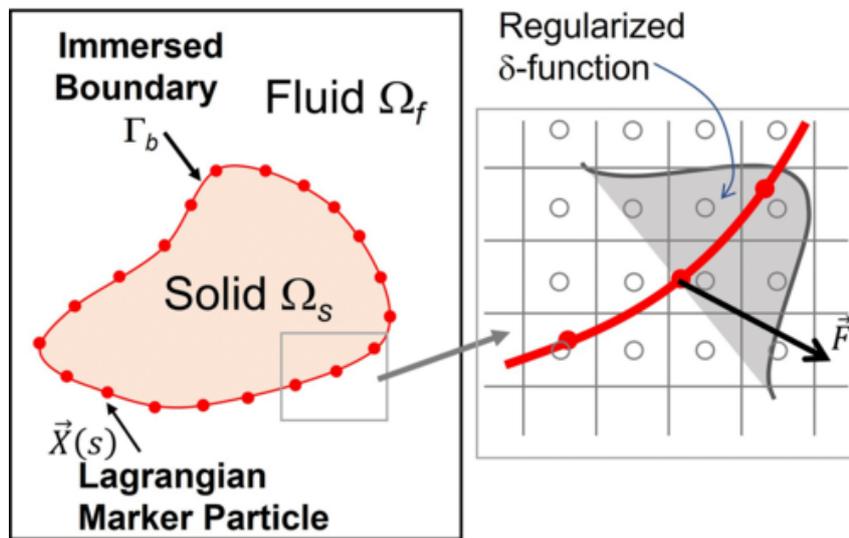


Figura 2.7: Schema del flusso incompressibile all'interno di un dominio 2D contenente un *immersed boundary* [19].

³L'interazione fluido struttura è un fenomeno multifisico che avviene in un sistema dove il flusso di un fluido provoca la deformazione di una struttura solida o viceversa.

Successivamente considerò la superficie come costituita da una serie di punti Lagrangiani (Figura 2.7) la cui posizione di ognuno è data dal vettore $\vec{X}(s, t)$ e la velocità equivale a quella locale, $\partial\vec{X}/\partial t = \vec{u}(\vec{X}, t)$: Collegò ognuno di questi punti con delle fibre elastiche prive di massa che si deformano con il flusso generando una tensione T data dalla legge di Hooke, nella forma:

$$T = \epsilon(|\partial\vec{X}/\partial s|; s, t) \quad (2.12)$$

ove ϵ è la deformazione locale nella fibra. L'effetto della fibra sul fluido è stato imposto trasmettendo una tensione data da:

$$\vec{F}(s, t) = \partial(T\hat{\tau})/\partial s \quad (2.13)$$

ove $\hat{\tau}$ è il versore tangente alla fibra.

Inoltre, poiché la griglia Cartesiana non coincide con i punti Lagrangiani della superficie, era necessario un metodo per trasferire le quantità come velocità del fluido e la forza della fibra tra i due insiemi di punti. Peskin utilizzò la funzione delta di Dirac per fare ciò e, in particolare, regolarizzò quest'ultima su una lunghezza finita dell'*immersed boundary*. Nella seguente equazione si riporta il trasferimento della forza esercitata dalla fibra dai punti Lagrangiani al campo Euleriano:

$$\vec{f}(\vec{x}, t) = \sum_k \vec{F}_k D(|\vec{x}_{i,j} - \vec{X}_k|) \quad (2.14)$$

ove D è la funzione di Dirac regolarizzata, mentre \vec{X}_k è il vettore posizione dei punti Lagrangiani, $\vec{x}_{i,j}$ il vettore posizione dei punti della griglia, \vec{F}_k è la forza esercitata dalle fibre.

Infine, in questo metodo Peskin non applicò la condizione al contorno esattamente nella posizione dell'*immersed boundary*, ma su una regione che si estende su diversi punti della griglia; ciò comportava che le condizioni al contorno di *no-slip* e *no-penetration* venissero soddisfatte solamente se la griglia avesse una risoluzione abbastanza elevata. In Figura 2.8 è possibile vedere un'applicazione di questo metodo presente nella tesi di Peskin.

Le differenze che esistono tra le varie tecniche implementate con questo tipo di frontiera stanno proprio nella scelta della funzione di Dirac regolarizzata D , anche detta *distribution function*: In Figura 2.9 vengono riportate varie funzioni sviluppate in studi diversi, ad

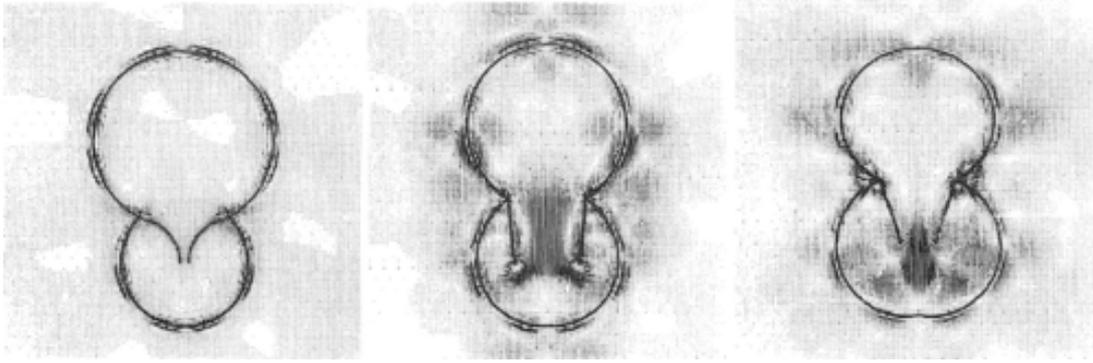


Figura 2.8: Risultati della simulazione del flusso all'interno del cuore [22].

esempio Beyer e Leveque [1] dimostrarono che è possibile ottenere una funzione D che preserva l'accuratezza dello schema spaziale.

Rigid Boundaries

Il metodo precedente è naturalmente adatto per i corpi elastici ma la sua applicazione ai flussi con corpi rigidi presenta problemi poiché le leggi costitutive utilizzate per le frontiere elastiche non sono generalmente ben poste nel limite rigido. Un primo approccio che si utilizza in presenza di *rigid boundaries* è chiamato *virtual boundary method*, usato da Goldstein [8]: L'idea è quella di trattare la superficie del corpo come un contorno virtualmente esistente inglobato nel fluido; questo contorno applica una forza sul fluido in modo che esso sia fermo sulla superficie (condizione di adesione, o *no-slip*) e questa vale:

$$\vec{F}(t) = \alpha \int_0^t \vec{u}(\tau) d\tau + \beta \vec{u}(t) \quad (2.15)$$

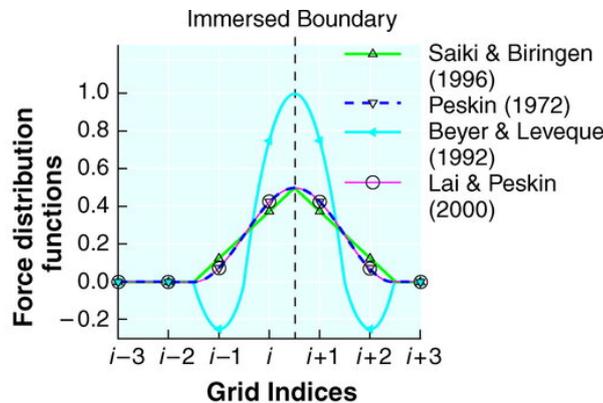


Figura 2.9: *Distribution functions* applicate in vari studi [18].

ove i coefficienti α e β sono arbitrari e possono essere scelti in modo da rispettare al meglio le condizioni al contorno, la velocità $\vec{u}(t)$ è quella dei punti sulla superficie.

Un altro approccio, il quale può essere visto come un caso specifico del *virtual boundary method*, è quello sviluppato da Lai [13]. Al fine di simulare un flusso attorno ad un *rigid boundary* con il metodo *immersed boundary* è necessario consentire alla frontiera di muoversi leggermente anziché rimanere fissa. Si introduce, quindi, una forza $\vec{F}(t)$ per assicurarsi che i punti della frontiera rimangano vicini alla superficie del corpo:

$$\vec{F}(t) = \kappa \left(\vec{X}(t) - \vec{X}^e(t) \right) \quad (2.16)$$

ove κ è una costante elastica positiva. L'interpretazione dell'equazione 2.16 è che i punti del contorno \vec{X} sono collegati a dei punti di equilibrio fissi \vec{X}^e tramite una molla molto rigida, la cui costante elastica vale κ . Quindi, se i punti del contorno si muovono dal luogo desiderato la forza della molla farà tornare questi punti indietro.

Considerazioni Generali

Un vantaggio del *continuous forcing approach* è che i metodi descritti sopra sono indipendenti dalla discretizzazione spaziale, a differenza dei metodi basati su un approccio di forzatura discreta. Pertanto, questo approccio può essere implementato in un risolutore Navier-Stokes esistente con relativa facilità. Uno svantaggio di questi metodi è che la smussatura della funzione di forzatura porta intrinsecamente a una incapacità di fornire una rappresentazione nitida dell'*immersed boundary* e quindi questi metodi non sono utili per flussi ad alto numero di Reynolds. Un altro svantaggio dell'approccio continuo è che tutti richiedono la soluzione delle equazioni governanti all'interno del corpo immerso.

2.2.2 Discrete Forcing Approach

Come già detto in precedenza, nei metodi che utilizzano il *discrete forcing approach* il termine forzante per l'imposizione della condizione al contorno viene aggiunto all'equazione di governo dopo che questa viene discretizzata. Anche questo secondo approccio è suddivisibile in due categorie: *metodi diretti* e *metodi indiretti*. Il vantaggio di questi approcci è l'assenza di costanti libere scelte dall'utente e l'eliminazione dei vincoli di stabilità.

Metodi Indiretti

Al netto di alcuni casi particolari, per le equazioni di Navier-Stokes non è possibile derivare analiticamente una funzione che permetta di rispettare le condizioni al contorno imposte dal problema: per questa ragione sono stati sviluppati alcuni metodi che riescono a costruire la forzante tramite alcune stime fatte a priori.

Il primo passo è la risoluzione delle equazioni senza il rispetto delle condizioni al contorno, ovvero risolvere la seguente equazione

$$[L]\{U^*\} = 0 \quad (2.17)$$

ad ogni *time step* per ottenere la velocità $\{U^*\}$ che rappresenta una soluzione predittiva.

A questo punto la forzante è ricavabile da:

$$\{\vec{f}_b\} \approx \{\vec{r}\} + [L]\{U^*\} - [L']\{U^*\} = \{\vec{r}\} - [L']\{U^*\} \quad (2.18)$$

ove $\{\vec{r}\} = \{\vec{U}_\Gamma - U^*\}d(|\vec{X}_k - \vec{x}_{i,j}|)$.

Metodi Diretti

Sebbene l'applicazione dei metodi IB a flussi a basso e moderato numero di Reynolds abbia avuto successo, la loro estensione a numeri di Reynolds più alti è impegnativa a causa della necessità di risolvere con precisione gli strati limite su superfici non allineate con le linee della griglia. In tali casi, l'accuratezza locale della soluzione assume maggiore importanza e la diffusione dell'effetto dell'IB introdotta dalla funzione di distribuzione della forza liscia è meno desiderabile. Per questo motivo, si possono considerare altri approcci in cui il confine immerso è mantenuto come un'interfaccia "netta" senza diffusione e dove viene posto maggiore enfasi sull'accuratezza locale vicino all'IB. Questo può essere generalmente

realizzato modificando lo *stencil* computazionale vicino al confine immerso per imporre direttamente la condizione al contorno sull'IB. Qui descriviamo due metodi che rientrano in questa categoria.

Approccio Ghost-Cell

Il metodo *ghost-cell* implementa alcune celle fantasma per imporre le condizioni al contorno sull'*immersed boundary*. Le celle fantasma sono definite come quelle celle della matrice solida che possiedono almeno una cella vicina che sia compresa nel fluido (Figura 2.10). Per ogni cella fantasma, si sviluppa uno schema di interpolazione che incorpora implicitamente la condizione al contorno sull'IB. Un'opzione semplice per costruire lo schema è l'interpolazione bilineare (o trilineare in tre dimensioni) ove una generica variabile del fluido ϕ può essere espressa nel seguente modo:

$$\phi = C_1x_1x_2 + C_2x_1 + C_3x_2 + C_4 \quad (2.19)$$

I quattro coefficienti dell'equazione possono essere calcolati per mezzo dei valori di ϕ nei *fluid nodes* F_1, F_2 e F_3 e al *boundary point* B_2 (Vedere Figura 2.10).

Applicare una ricostruzione lineare è accettabile per flussi laminari oppure per flussi ad alto numero di Reynolds che hanno il primo punto della griglia localizzato nel *viscous sublayer*. Tuttavia, a elevati numeri di Reynolds, quando la risoluzione è marginale, la

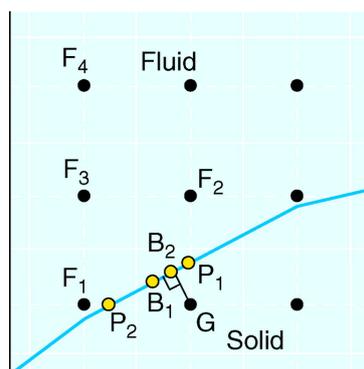


Figura 2.10: Rappresentazione dei punti in prossimità dell'*immersed boundary* usati nell'approccio *ghost-cell*. F_i sono i *fluid points*, G è il *ghost point* (ovvero quello della cella fantasma) e B_i e P_i sono i punti cui le condizioni al contorno possono essere applicate [18].

ricostruzione lineare potrebbe portare a previsioni errate. Per questa ragione si utilizzano metodi più sofisticati, ad esempio si può eseguire una interpolazione lineare in direzione tangenziale t e quadratica lungo la normale n :

$$\phi = C_1 n^2 + C_2 n t + C_3 n + C_4 t + C_5 \quad (2.20)$$

Anche in questo caso si possono ricavare i valori dei coefficienti usando i *fluid nodes* e i *boundary points*.

Indipendentemente dal particolare schema di interpolazione utilizzato, il valore della variabile al nodo della cella fantasma, ϕ_G , può essere espresso come:

$$\phi_G = \sum \omega_i \phi_i \quad (2.21)$$

La sommatoria si estende su tutti i punti dello stencil, inclusi uno o più punti di confine, e ω_i sono dei coefficienti noti dipendenti dalla geometria. Risolvendo questa equazione simultaneamente con le equazioni di Navier-Stokes discretizzate per i nodi del fluido viene risolto il problema.

Approccio Cut-Cell

Il *cut-cell method*, il quale fa sempre parte della categoria dei metodi diretti, fu proposto per la prima volta da Ye *et al.* [32] nel 1999 con l'obiettivo di far rispettare le leggi di conservazione anche nelle vicinanze dell'*immersed boundary* tramite un approccio ai volumi finiti. In questo metodo, le celle nella griglia cartesiana che risultano "tagliate" dall'IB vengono identificate: Se il centro della *cut-cell* fa parte del dominio fluido, questa cella viene rimodellata, eliminando la porzione che si trova nel solido; se, invece, il centro cade nella parte solida la cella viene assorbita completamente dalle celle vicine al fine di evitare successivi problemi di stabilità. Il risultato di questa operazione è l'ottenimento dei volumi di controllo, i quali hanno una forma tipicamente trapezoidale, come mostrato in Figura 2.11.

L'approccio proposto da Ye *et al.* è quello di esprimere, similmente all'approccio *ghost-cell*, una variabile del flusso ϕ in termini di una funzione interpolante in una regione appropriata e valutare i flussi f basandosi su questa funzione. Ad esempio, al fine di valutare il flusso nella faccia sud-ovest f_{sw} (regione trapezoidale in Figura 2.11), ϕ viene

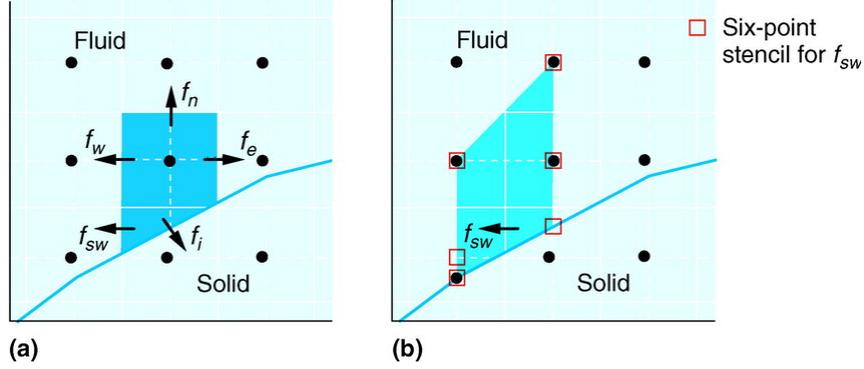


Figura 2.11: Metodo *cut-cell*. (a) Volume finito trapezoidale formato nelle vicinanze dell'IB ove f rappresenta il flusso di una generica variabile attraverso le rispettive facce. (b) Regione di interpolazione e stencil sviluppato per l'approssimazione del flusso f_{sw} . [18].

espressa come una funzione lineare in x e quadratica in y :

$$\phi = c_1xy^2 + c_2y^2 + c_3xy + c_4y + c_5x + c_6 \quad (2.22)$$

ove sono presenti sei coefficienti incogniti. Questa equazione rappresenta il modo più compatto per valutare la variabile ϕ nella posizione sud-ovest almeno al secondo ordine. Un approccio simile può essere impiegato per valutare il flusso sulla faccia est f_e così come il flusso di interfaccia f_i . Questo approccio porta a uno schema di discretizzazione che è accurato al secondo ordine sia globalmente che localmente e soddisfa la conservazione di massa e quantità di moto indipendentemente dalla risoluzione della griglia.

Considerazioni Generali

Tutti i metodi che implementano un approccio *discrete forcing*, non avendo una funzione δ di Dirac e, di conseguenza, non necessitando di una sua regolarizzazione, eliminano il compromesso tra accuratezza e stabilità numerica richiesto dei metodi *continuous forcing* e questo consente di raggiungere più facilmente un'accuratezza superiore al primo ordine. Inoltre questi metodi consentono di ottenere una rappresentazione accurata dell'IB e questo è desiderabile, specialmente a numeri di Reynolds più elevati. Uno svantaggio di questi metodi è che dipendono fortemente dal metodo di discretizzazione, consentendo, tuttavia, un controllo diretto sull'accuratezza numerica, sulla stabilità e sulle proprietà di conservazione del risolutore. Un altro svantaggio è che questi metodi non sono immediati

da implementare a causa della discretizzazione iniziale e dell'introduzione successiva di un termine di forzatura.

Per quanto riguarda il metodo *cut-cell*, i vantaggi riguardano la possibilità di imporre accuratamente le condizioni al contorno sul corpo e il fatto di basarsi su volumi finiti, quindi la rigorosa conservazione di massa e quantità di moto è garantita anche nelle vicinanze del confine immerso. Uno degli svantaggi di questo approccio è il seguente. Implementare le condizioni al contorno in celle irregolari richiede un gran numero di trattamenti speciali, il che potrebbe comportare complessità nella codificazione. Quando si utilizzano le celle tagliate, queste celle non dovrebbero diventare troppo piccole altrimenti ciò potrebbe non solo causare problemi di stabilità, ma anche rallentare la convergenza del risolutore di Poisson. Infine, potrebbe essere necessaria una procedura di soluzione iterativa a causa dello *stencil* irregolare vicino al confine immerso.

Capitolo 3

Signed Distance Field

L'obiettivo principale di questa tesi è quello di implementare un *Signed Distance Field* (SDF) per svolgere simulazioni fluidodinamiche che si basano su tecniche IBM.

Nel nostro caso un SDF viene generato a partire da una geometria in formato STL. In questo capitolo verrà fornita una breve descrizione di tale formato, seguita da un'analisi approfondita delle proprietà di SDF e degli algoritmi sviluppati per la generazione di un SDF a partire da una geometria in formato STL. L'incorporazione di SDF all'interno del codice combinato alla tecnica IBM sarà argomento di discussione del paragrafo 5.3

3.1 Formato STL

STL è l'acronimo di *STereo Lithography* ed è un formato di file, binario o ASCII, nato per i software di stereolitografia CAD nel 1987. Questo formato approssima le superfici tridimensionali di un modello solido con triangoli orientati (detti *facets*) di diverse dimensioni e forme al fine di ottenere una rappresentazione sufficientemente uniforme e, quindi, adatta all'utilizzo delle macchine.

Il formato STL prevede che venga generata una mesh triangolare tramite una tassellatura della superficie solida: per applicazioni ingegneristiche la mesh deve anche contenere l'informazione riguardo quale parte del triangolo contiene l'interno del solido [26]. Il contenuto del file STL è quindi un set di triangoli orientati, ognuno descritto dai seguenti dati numerici (Figura 3.1):

$$\begin{array}{ccc}
n_x^{(i)} & n_y^{(i)} & n_z^{(i)} \\
p_{1x}^{(i)} & p_{1y}^{(i)} & p_{1z}^{(i)} \\
p_{2x}^{(i)} & p_{2y}^{(i)} & p_{2z}^{(i)} \\
p_{3x}^{(i)} & p_{3y}^{(i)} & p_{3z}^{(i)}
\end{array}$$

ove $(n_x^{(i)}, n_y^{(i)}, n_z^{(i)})$ sono le coordinate della normale uscente della i -esima *facet* e $(p_{kx}^{(i)}, p_{ky}^{(i)}, p_{kz}^{(i)})$ sono le coordinate del k -esimo vertice contenuto nell' i -esimo triangolo, con $i = 1, \dots, N$ ove N è il numero di triangoli complessivo della mesh.

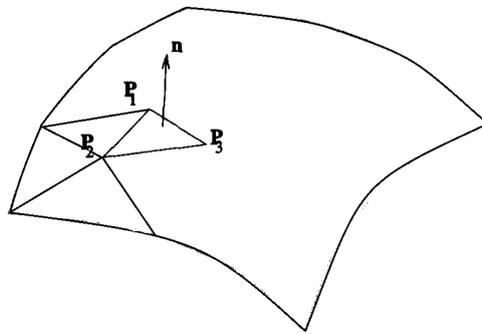


Figura 3.1: *Facet* della superficie nel formato STL [26].

In Figura 3.2 viene riportato un esempio di tassellatura di una superficie in formato STL: nei casi che si andranno a studiare in seguito verrà svolta questa operazione prima di calcolare il SDF della geometria da usare per la simulazione.

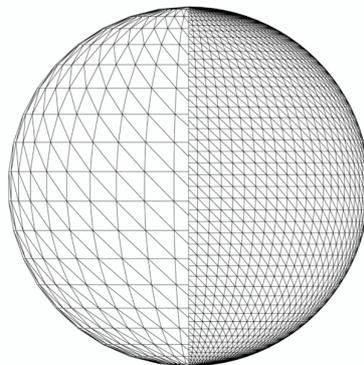


Figura 3.2: Tassellatura di una geometria.

3.2 SDF: Definizione e Proprietà

Il *Signed Distance Field* è un campo scalare ¹ generato da una superficie S tale che, per ogni punto dello spazio \vec{p} , il valore assoluto del campo $|\phi(\vec{p})|$ è la più piccola distanza tra \vec{p} e S . Formalmente si scrive:

$$|\phi(\vec{p})| = \min_{\vec{q} \in S} \|\vec{p} - \vec{q}\|_2 \quad (3.1)$$

ove $\|\cdot\|_2$ denota la norma Euclidea. Per convenzione il segno di $\phi(\vec{p})$ è positivo quando \vec{p} è al di fuori di S , è negativo quando \vec{p} è all'interno di S , mentre il valore è nullo quando il punto giace sulla superficie [34].

Si possono ottenere delle espressioni analitiche per il SDF per alcune superfici elementari. Ad esempio il SDF di un cerchio con raggio unitario centrato nell'origine vale:

$$\phi = \sqrt{x^2 + y^2} - 1 \quad (3.2)$$

In Figura 3.3 sono riportati i SDF per il cerchio e anche per il rettangolo di dimensioni $2r_a \times 2r_b$

Una caratteristica distintiva del SDF è che i suoi contorni hanno forme molto simili alla superficie presentandosi come dilatazioni o restringimenti di essa che, a sua volta, è il contorno per $\phi = 0$. Quindi una caratteristica peculiare dei SDF è quella non solo di codificare la superficie ma anche di trasmetterla all'intero spazio: questo conferisce molte proprietà utili per calcoli geometrici. In Tabella 3.1 sono riportate alcune proprietà del SDF: ad esempio in Figura 3.4 è rappresentato il SDF dato dall'unione tra quello del cerchio e quello del rettangolo, per ottenerlo è bastato prendere il minimo tra i due campi.

¹In matematica e fisica un campo scalare è una funzione che associa uno scalare a ogni punto di uno spazio.

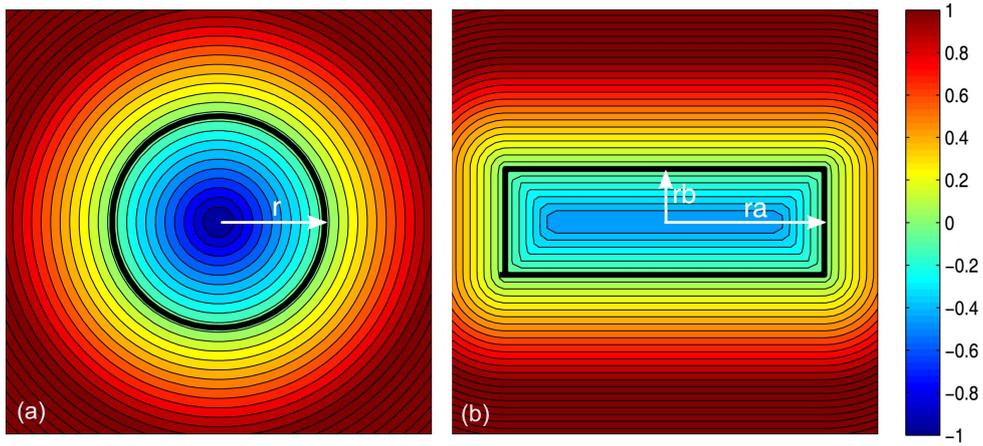


Figura 3.3: *Signed Distance Field* di un cerchio di raggio unitario e di un rettangolo di lati $2r_a \times 2r_b$ [34].

Proprietà Geometriche	
$\phi(\vec{p})$	Minima distanza del punto \vec{p} dalla superficie
$\nabla\phi(\vec{p})$	Versore normale al contorno della SDF
$\nabla^2\phi(\vec{p})$	Curvatura al punto \vec{p}
Proprietà Booleane	
$\min(\phi_1, \phi_2)$	SDF dell'unione di S_1 e S_2
$\max(\phi_1, \phi_2)$	SDF dell'intersezione tra S_1 e S_2
$\max(\phi_1, -\phi_2)$	SDF della sottrazione tra S_1 e S_2

Tabella 3.1: Lista di proprietà del SDF.

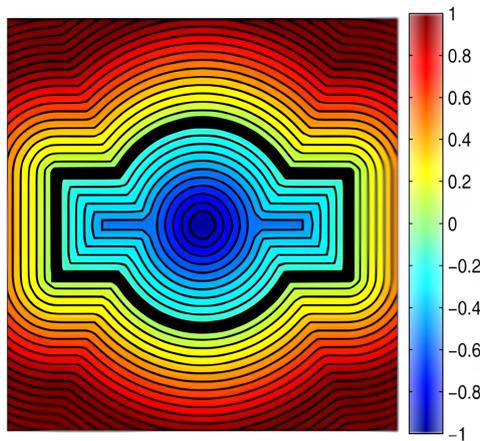


Figura 3.4: Signed Distance Field data dall'unione dei due precedenti [34].

3.3 Modello DeepSDF e modulo *mesh-to-sdf*

Il codice sviluppato in questa tesi utilizza il modulo *mesh-to-sdf* per il calcolo di SDF a partire da una geometria in formato STL. A sua volta questo modulo si basa sul modello *DeepSDF*, ovvero un *learned continuous SDF* che consente una rappresentazione ad alta qualità a partire da dati in input. La scelta nell'usare questo modello è dovuta al fatto che, a differenza di altri metodi per la generazione del SDF, non si utilizza una griglia regolare per la discretizzazione della SDF perché viene invece prodotto un campo continuo. In Figura 3.5 viene riportato un esempio di utilizzo di questo modello ove viene mostrato come il SDF venga generato a partire dal contorno della geometria.

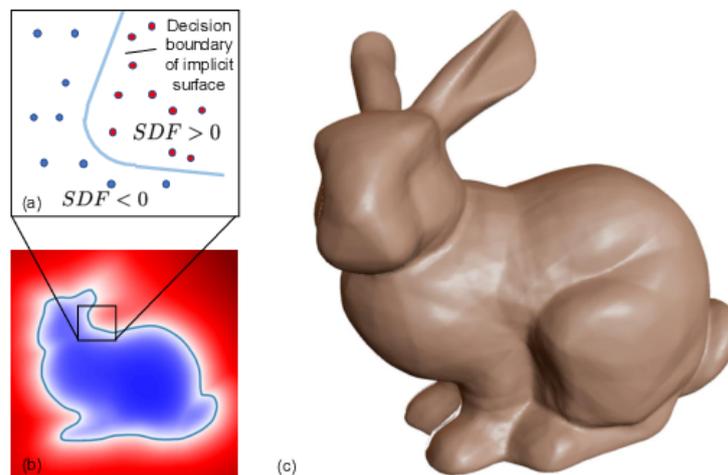


Figura 3.5: Rappresentazione DeepSDF applicata allo *Stanford Bunny*: (a) Le tre zone del campo, $SDF = 0$ rappresenta il contorno della superficie, $SDF < 0$ i punti all'interno, mentre $SDF > 0$ i punti all'esterno, (b) Rappresentazione del SDF in due dimensioni, (c) Superficie 3D ottenuta per $SDF = 0$ [21].

3.3.1 Modulo *mesh-to-sdf*

La *repository mesh-to-sdf* [10], basandosi sul modello *DeepSDF*, implementa la seguente procedura per il calcolo di SDF:

1. Data una geometria STL viene eseguita una scansione della superficie attraverso 100 laser a diverse angolazioni, ricavando il *normal buffer*, matrice che memorizza le informazioni delle normali delle superfici, e il *depth buffer*, matrice che memorizza le distanze tra laser e superficie.
2. Per ogni scansione vengono svolte delle operazioni sulle matrici, ottenendo una nuvola di punti come risultato.
3. Per una serie di punti scelti, chiamati *query points*, viene individuata la più piccola distanza dalla superficie, determinando il valore di SDF.
4. Il segno di SDF viene determinato attraverso dei metodi precisi a seconda dei casi. Ad esempio è possibile utilizzare le normali calcolate nei punti più vicini alla superficie, calcolando il prodotto scalare di questi.

In Figura 3.6 viene riportato l'esempio presentato dalla *repository* ove per la geometria considerata, in questo caso il modello STL, viene calcolato il SDF per una serie di *query points*, i quali vengono raffigurati con colore diverso a seconda della distanza dalla superficie.

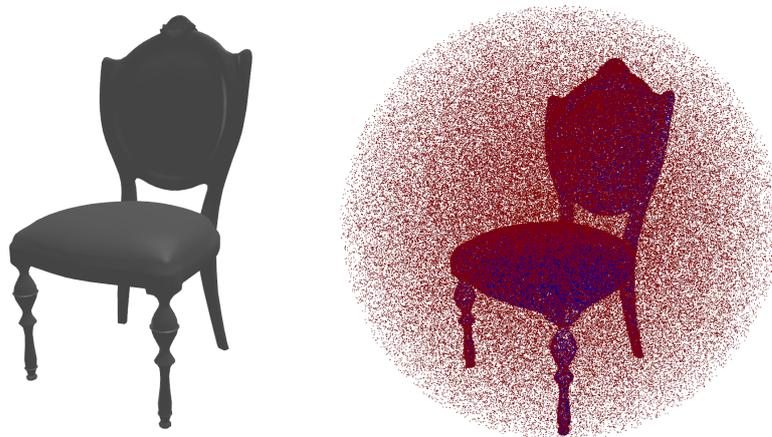


Figura 3.6: A sinistra la geometria STL; a destra i *query points* per i quali viene calcolato il *signed distance field* [10].

Capitolo 4

Flussi Turbolenti e Modello LES

La maggior parte dei flussi che si incontrano in pratiche di ingegneria e in natura sono turbolenti, questo vale anche per i casi che verranno analizzati in questo lavoro di tesi. In generale si identifica un flusso turbolento come il moto non stazionario, irregolare e caotico di un fluido.

Per questa ragione, in questo capitolo viene presentata una breve descrizione del fenomeno della turbolenza per poi discutere del modello *Large Eddy Simulation*, tecnica numerica utilizzata nel codice sviluppato per simulare i flussi in regime turbolento.

4.1 Turbolenza

In generale un flusso turbolento possiede le seguenti caratteristiche [12]:

1. *Caoticità*: I flussi turbolenti sono intrinsecamente imprevedibili. Anche piccoli cambiamenti nelle condizioni iniziali possono portare a differenze significative nel comportamento del fluido, rendendo difficile la loro modellizzazione e previsione.
2. *Non Linearità*: I flussi turbolenti sono estremamente non lineari.
3. *Diffusività*: A causa della mescolanza macroscopica delle particelle fluide, i flussi turbolenti sono caratterizzati da un rapido tasso di diffusione della quantità di moto e del calore.
4. *Vorticità*: La turbolenza è caratterizzata da alti livelli di vorticità fluttuante e una caratteristica distintiva della turbolenza è l'esistenza di un'enorme gamma di

dimensioni dei vortici. L'energia viene trasmessa dai grandi vortici ai piccoli vortici tramite interazioni non lineari, fino a quando viene dissipata dalla diffusione viscosa nei vortici più piccoli, la cui dimensione è dell'ordine dei millimetri.

5. *Dissipazione*: Il meccanismo di allungamento dei vortici trasferisce energia e vorticità a scale sempre più piccole, fino a quando i gradienti diventano così grandi che vengono dissipati dalla viscosità. I flussi turbolenti richiedono quindi una fornitura continua di energia per compensare le perdite viscosive.

In generale per distinguere un flusso turbolento da uno laminare si utilizza il numero di Reynolds Re , definito come:

$$Re = \frac{\rho u L}{\mu} = \frac{u L}{\nu} \quad (4.1)$$

Il numero di Reynolds descrive il rapporto tra le forze inerziali e le forze viscosive: considerando un flusso turbolento le prime dominano sulle seconde, mentre per un flusso laminare vale il contrario.

4.1.1 Quantità Medie

Le variabili di un flusso turbolento non sono deterministiche e devono essere trattate come variabili stocastiche o *random*. Data $u(t)$ una qualsiasi variabile misurata di un flusso turbolento vengono considerati due casi: nel primo le caratteristiche medie di $u(t)$ non variano nel tempo (caso stazionario), mentre nel secondo avviene il contrario (caso non stazionario). Per il caso stazionario è sempre possibile definire il valore medio della quantità $u(t)$ nel tempo:

$$\bar{u} \equiv \lim_{t_0 \rightarrow \infty} \frac{1}{t_0} \int_0^{t_0} u(t) dt. \quad (4.2)$$

Nel caso non stazionario, invece, le caratteristiche medie variano con il tempo: un esempio pratico è dato dalla velocità di un jet nel momento in cui la pressione nel serbatoio cala. In questo caso non è possibile utilizzare l'equazione 4.2 poiché non è specificato quanto esteso sia l'intervallo $[0, t_0]$ che si considera per il calcolo delle medie. In casi come questo bisogna eseguire un gran numero di esperimenti condotti nelle stesse condizioni ottenendo un *ensemble*, ovvero un insieme di dati. Si può quindi calcolare la *ensemble average* (o media alla Reynolds) della quantità u al tempo t , ottenendo il valore medio di una

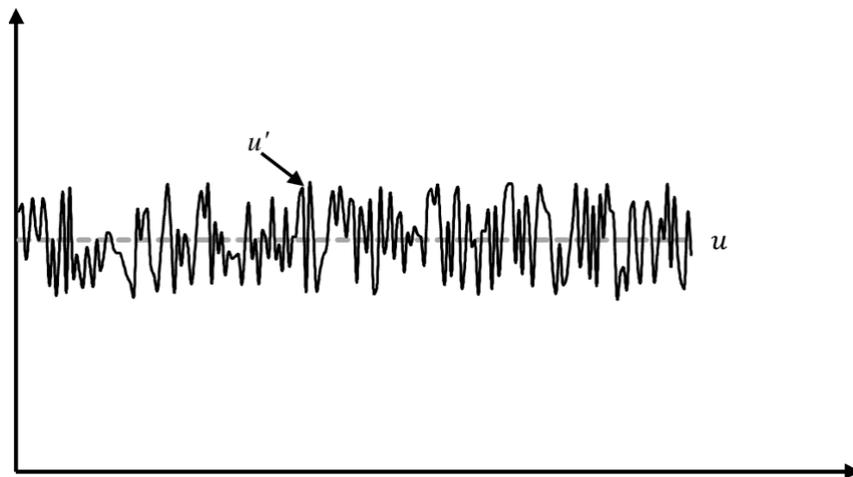


Figura 4.2: Rappresentazione della quantità fluttuante u' rispetto al valore medio [12].

In generale le strutture vorticoshe di taglia più grande avranno una dimensione ℓ_0 comparabile con quella dell'oggetto che le ha originate $\ell_0 \approx L_0$, la loro velocità caratteristica $u(\ell_0)$ darà simile a quella U del campo complessivo. Il numero di Reynolds di queste scale sarà grande e comparabile a quello del flusso complessivo, quindi gli effetti diretti della viscosità sono piccoli.

Secondo la teoria di Richardson gli *eddies* di grandi dimensioni, i quali contengono la maggior parte dell'energia cinetica estratta dal flusso medio, sono instabili e tendono a distruggersi trasferendo la propria energia agli *eddies* più piccoli. Questi, a loro volta, distruggendosi cedono la loro energia agli *eddies* di dimensioni ancora più piccole. Il processo descritto si ripete incessantemente fino a che $Re(\ell) = \frac{u(\ell)\ell}{\nu}$ non assume valori sufficientemente piccoli da rendere stabile la dinamica degli *eddies* permettendo, così, alla viscosità di dissipare l'energia fin qui trasferita. Richardson riassume così questo fenomeno:

*Big whorls have little whorls,
Which feed on their velocity;
And little whorls have lesser whorls,
And so on to viscosity
(in the molecular sense).*

La dissipazione è quindi posta alla fine del processo e quindi l'energia da dissipare ϵ dipende dalle strutture vorticoshe più grandi. Questi *eddies* possiedono un'energia dell'ordine di

u_0^2 e una scala temporale di $\tau_0 = \frac{\ell_0}{u_0}$ quindi il trasferimento di energia scala con:

$$\epsilon \sim \frac{u_0^2}{\tau_0} \sim \frac{u_0^3}{\ell_0} \quad (4.5)$$

Quindi l'energia da dissipare dipende dalle strutture vorticose di grandi scale ma risulta indipendente dalla viscosità ν . Al fine di investigare la dimensione degli *eddies* più piccoli responsabili della dissipazione di energia risulta necessario introdurre la teoria di Kolmogorov.

4.1.3 Teoria di Kolmogorov

Secondo la teoria di Kolmogorov [11] il flusso macroscopicamente risulta dipendente dalla geometria, mentre a livello microscopico il flusso risulta universale e quindi non risente più dell'influenza della geometria che l'ha generato. La teoria si struttura in tre ipotesi. La prima ipotesi riguarda l'isotropia dei moti su piccola scala. In generale, i grandi vortici sono anisotropi e sono influenzati dalle condizioni al contorno del flusso. Kolmogorov sostenne che le distorsioni direzionali delle grandi scale vengono perse nel processo caotico di riduzione della scala, mediante il quale l'energia viene trasferita a vortici successivamente sempre più piccoli. Pertanto si enuncia:

Ipotesi zero: A un numero di Reynolds sufficientemente alto, i moti turbolenti su piccola scala $\ell \ll \ell_0$ sono statisticamente isotropi.

Risulta utile ora introdurre una scala $\ell_{LI} \sim \frac{1}{6}\ell_0$ come linea di demarcazione tra i grandi vortici anisotropi ($\ell > \ell_{LI}$) e i piccoli vortici isotropi ($\ell < \ell_{LI}$). Così come l'informazione direzionale delle grandi scale viene persa mentre l'energia passa attraverso la cascata, Kolmogorov sostenne che anche tutte le informazioni riguardanti la geometria dei grandi vortici – determinate dal campo di flusso medio e dalle condizioni al contorno – vengono perse. Di conseguenza, le statistiche dei moti su piccola scala sono in un certo senso universali – simili in ogni flusso turbolento ad alto numero di Reynolds. Nella seconda ipotesi, Kolmogorov ipotizza da quali quantità i moti su piccola scala dipendono, enunciando:

Prima ipotesi di similarità: In ogni flusso turbolento a un numero di Reynolds sufficientemente alto, le statistiche dei moti su piccola scala ($\ell < \ell_{LI}$) hanno una forma universale che è determinata unicamente da ν e ϵ .

Eseguendo un'analisi dimensionale si ottengono le grandezze di lunghezza caratteristica, velocità caratteristica e tempo caratteristico alla scala di Kolmogorov:

$$\eta \equiv \left(\frac{\nu^3}{\epsilon} \right)^{\frac{1}{4}} \quad (4.6)$$

$$u_\eta \equiv (\epsilon \nu)^{\frac{1}{4}} \quad (4.7)$$

$$\tau_\eta \equiv \left(\frac{\nu}{\epsilon} \right)^{\frac{1}{2}} \quad (4.8)$$

Questa scala risulta la più piccola possibile ed è quindi proprio la scala dissipativa. Infatti, calcolando il numero di Reynolds, esso risulta essere unitario, in accordo col fatto che su scale sempre più piccole Re deve essere sufficientemente piccolo per far accadere la dissipazione. Inoltre vale:

$$\epsilon_\eta = \nu \left(\frac{u_\eta}{\eta} \right)^2 = \epsilon \quad (4.9)$$

mostrando che l'energia di dissipazione eguaglia quella della scala dissipativa.

Calcolando il rapporto tra le quantità della scala dissipativa e di quelle delle strutture più grandi si ottiene:

$$\eta/\ell_0 \sim Re^{-3/4} \quad (4.10)$$

$$u_\eta/u_0 \sim Re^{-1/4} \quad (4.11)$$

$$\tau_\eta/\tau_0 \sim Re^{-1/2} \quad (4.12)$$

Da queste relazioni si può vedere che per alti numeri di Reynolds, la velocità e le scale temporali dei vortici della scala dissipativa sono piccoli in confronto alle grandi strutture vorticose; inoltre si può dedurre che, per un Re sufficientemente elevato, esista una scala di *eddies* ℓ piccola rispetto a ℓ_0 e grande rispetto a η tale per cui il numero di Reynolds della scala è elevato e, di conseguenza, il moto è affetto in piccola parte dalla viscosità. Da questa considerazione viene formulata l'ultima ipotesi, ovvero:

Seconda ipotesi di similarità: In ogni flusso turbolento a un numero di Reynolds sufficientemente elevato, le statistiche dei moti di scala ℓ nell'intervallo $\ell_0 \gg \ell \gg \eta$ assumono una forma universale determinata unicamente da ϵ , indipendentemente da ν .

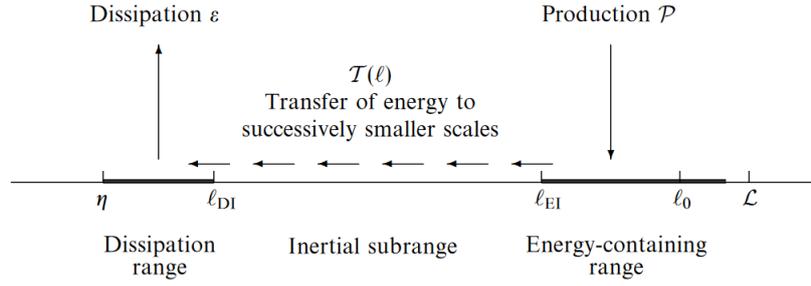


Figura 4.3: Schema della cascata di energia per numeri di Reynolds elevati [23].

Queste scale vengono definite inerziali ed è possibile calcolare velocità e tempo caratteristico:

$$u(\ell) = (\epsilon \ell)^{1/3} = u_\eta \left(\frac{\ell}{\eta} \right)^{1/3} \sim u_0 \left(\frac{\ell}{\ell_0} \right)^{1/3} \quad (4.13)$$

$$\tau(\ell) = \left(\frac{\ell^2}{\epsilon} \right)^{1/3} = \tau_\eta \left(\frac{\ell}{\eta} \right)^{2/3} \sim \tau_0 \left(\frac{\ell}{\ell_0} \right)^{2/3} \quad (4.14)$$

A queste scale l'energia viene trasferita per alimentare i vortici più piccoli e infatti vale:

$$\epsilon_\ell = u(\ell)^2 / \tau(\ell) \sim \epsilon \quad (4.15)$$

Quindi, introducendo un'energia P , detta produzione, nelle scale più grandi, avviene un trasferimento inerziale dell'energia fino alle scale più piccole, per le quali avviene la dissipazione. Questo processo viene riportato in Figura 4.3.

4.2 Modellistica della Turbolenza

Nella sezione precedente si è potuto vedere come i flussi in regime turbolento siano un fenomeno complesso da studiare e, per questa ragione, sono stati sviluppati diversi approcci numerici per simulare questa tipologia di flussi [23]. Tra le principali tecniche vi sono le simulazioni dirette DNS (*Direct Numerical Simulation*) le quali risolvono completamente le equazioni di Navier-Stokes senza approssimazioni significative, catturando così tutte le scale spaziali e temporali del flusso. Tuttavia, a causa dell'enorme risoluzione computazionale richiesta, le DNS sono limitate a numeri di Reynolds relativamente bassi. Per numeri di Reynolds più elevati si utilizzano solitamente due tecniche: La prima implementa modelli di turbolenza RANS (*Reynolds Averaged Navier Stokes*), i quali applicano

l'operatore media visto in precedenza nelle equazioni di Navier Stokes e vengono utilizzati dei modelli di turbolenza per rappresentare gli effetti delle piccole scale non risolte; la seconda è il metodo LES (*Large Eddy Simulation*) che risolve le scale più grandi del flusso in modo diretto, mentre le scale più piccole vengono modellate.

Per i problemi che si andranno a trattare successivamente si utilizzerà il metodo LES poiché offre una combinazione di accuratezza nei dettagli delle grandi scale e flessibilità nell'applicazione, di conseguenza verrà studiato più nel dettaglio nel seguente paragrafo.

4.2.1 Modello LES

La tecnica LES è in partenza una tecnica più avanzata della RANS in quanto affonda le sue radici nelle due ipotesi fondamentali di similarità e universalità che sottostanno alla teoria K41 di Kolmogorov. L'idea di base della LES è quella di simulare le grandi scale che dipendono dal particolare flusso in esame e di far intervenire il modello per stimare l'effetto delle piccole scale sulle grandi; inoltre l'obiettivo della LES è quella di descrivere l'evoluzione temporale delle grandi scale e quindi simula anche i processi non stazionari che si riscontrano nei flussi turbolenti.

Ci sono quattro passaggi concettuali per effettuare una LES:

1. Un operazione di *filtraggio* al fine di decomporre la velocità $u(x, t)$ in una somma di una componente filtrata $\tilde{u}(x, t)$ e di una componente residua $u'(x, t)$. Il campo di velocità filtrato rappresenta il moto delle grandi strutture turbolente.
2. Le equazioni di evoluzione del campo di velocità filtrata derivano dalle equazioni di Navier Stokes. Queste equazioni sono nella forma standard, con l'equazione della quantità di moto che contiene il tensore degli sforzi residui.
3. Modellare il tensore degli sforzi residui con un modello *eddy-viscosity* per la chiusura del problema.
4. Risolvere numericamente le equazioni filtrate e modellizzate.

La velocità filtrata $\tilde{u}_i(x, t)$ viene definita attraverso l'operatore di filtraggio G_Δ :

$$\tilde{u}_i(x, t) = G_\Delta[u_j] = \int_V u_i(x - r, t) G_\Delta(r, x) dr \quad (4.16)$$

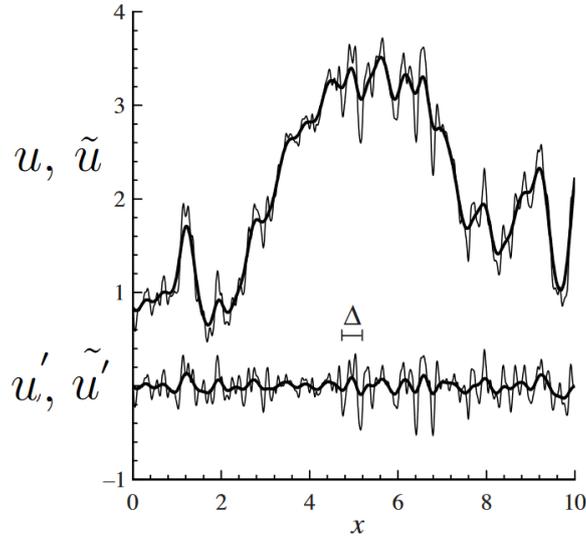


Figura 4.4: In alto velocità istantanea u e la relativa velocità \tilde{u} filtrata; in basso velocità residua u' e la relativa quantità filtrata \tilde{u}' .

ove l'integrazione è estesa in tutto il dominio fluido, Δ rappresenta l'ampiezza caratteristica del filtro considerato, ossia la lunghezza di riferimento al sotto della quale il filtro taglia le fluttuazioni. Il suo valore dovrà ricadere all'interno del range inerziale al fine di costruire un modello indipendente dal flusso simulato, dato che le scale più piccole hanno una dinamica universale. Infine x e r sono rispettivamente il punto in cui viene calcolata la grandezza filtrata e la distanza da tale punto. L'operatore di filtraggio rispetta la seguente condizione di normalizzazione:

$$\int_V G(r, x) dr = 1 \quad (4.17)$$

Definita la forma del filtro e la sua ampiezza possiamo ottenere la velocità filtrata \tilde{u}_i e risulta definita la decomposizione in velocità filtrata e residua (o di sottogriglia) u'_i :

$$u_i = \tilde{u}_i + u'_i \quad (4.18)$$

In Figura 4.4 viene riportato l'andamento delle varie velocità. Nella simulazione LES possono essere utilizzati sia filtri espliciti (come il filtro *box* o il filtro Gaussiano) oppure anche filtri impliciti ove Δ eguaglia il passo della griglia.

Per ottenere le equazioni che governano i campi di velocità filtrati bisogna applicare

il filtro alle equazioni di Navier-Stokes, ottenendo (in notazione indiciale):

$$\partial_t \tilde{u}_i + \partial_j (\tilde{u}_i \tilde{u}_j) = -\frac{1}{\rho} \partial_i \tilde{p} + \frac{1}{\rho} \partial_j \left(2\mu \tilde{E}_{ij} \right) - \partial_j \tau_{ij}^r \quad (4.19)$$

$$\partial_j \tilde{u}_j = 0 \quad (4.20)$$

nelle quali è stato definito il tensore degli sforzi residui (o di sottogriglia):

$$\tau_{ij}^r = u_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j \quad (4.21)$$

Questo tensore degli sforzi tiene conto solamente dell'effetto delle piccole scale non risolte sul campo evoluto e per risolvere le equazioni filtrate è necessario fornire una modellazione di questo tensore. Di seguito viene descritto il modello di Smagorinsky che fa uso del concetto di *eddy viscosity*.

Dividendo il tensore in parte sferica e deviatorica si può scrivere, considerando l'energia cinetica residua K^r :

$$\tau_{ij}^r = \frac{2}{3} K^r \delta_{ij} - 2\nu^r \tilde{E}_{ij} \quad (4.22)$$

e in tale espressione compare la viscosità cinematica residua o di sottogriglia $\nu^r = \nu^r(x, t)$ che dovrà essere modellata. Per trovare un modello per questa grandezza bisogna considerare che, dalle ipotesi di Kolmogorov, il flusso di energia che arriva alla scala Δ vale $\epsilon_\Delta^T = \frac{u_\Delta^3}{\Delta}$ e questo eguaglia il flusso dissipato ovvero $\epsilon_\eta = \nu \frac{u_\eta^2}{\eta^2}$. Poiché le scale più piccole di Δ , ove avviene la dissipazione, non vengono simulate si considera di dissipare tutta l'energia alla scala Δ , di conseguenza le due quantità possono essere eguagliate a meno di una costante C_S :

$$\nu_r = C_S \nu_\Delta \Delta \quad (4.23)$$

Infine, sostituendo alla velocità un valore approssimato si ottiene una stima chiusa della viscosità residua data da:

$$\nu^r = (C_S \Delta)^2 \tilde{S} \quad (4.24)$$

ove $S = \sqrt{2\tilde{E}_{ij}\tilde{E}_{ij}}$ e C_S ha un valore compreso tra 0.06 e 0.2. Poiché tutti i valori sono ora noti, il modello risulta chiuso.

4.3 Wall-Modeled LES

La presenza delle pareti nella simulazione del flusso di un condotto può essere problematica da risolvere per il modello LES: vicino alle pareti le strutture vorticosi scalano con la lunghezza viscosa $\delta_v = \nu/u_\tau$ invece che con lo spessore dello strato limite τ . Definendo il numero di Reynolds di attrito come il rapporto tra i due spessori,

$$Re_\tau = \delta/\delta_v$$

crescendo questo valore, i vortici vicino alla parete diminuiscono la loro dimensione relativamente allo spessore dello strato limite. Le simulazioni DNS e LES possono comunque essere usate ma, raggiungendo elevati numeri di Reynolds, esse diventano estremamente costose a livello computazionale. Per migliorare quindi i risultati del modello LES all'interno dello strato limite ci sono due metodi principali:

- Wall-Resolved LES (WRLES): I vortici vicini alla parete vengono *risolti* sulla griglia computazionale.
- Wall-Modeled LES (WMLES): I vortici vicini alla parete vengono *modellati* sulla griglia computazionale.

Uno studio di Choi e Moin [3] esprime una relazione tra costo computazionale del metodo (espresso in numero di punti della griglia richiesti) e numero di Reynolds:

$$N_{NDS} \sim Re^{37/14} \quad N_{WRLES} \sim Re^{13/7} \quad N_{WMLES} \sim Re$$

Il numero di punti necessari per il modello WMLES risulta essere simile al numero di Reynolds e ciò lo rende un'alternativa interessante, anche per i casi studiati nel presente lavoro di tesi. Usando questo approccio, al fine di replicare correttamente gli effetti dei vortici vicini a parete, va aggiunta una *eddy viscosity* v_a in prossimità delle pareti. Al fine di ottenere questa quantità viene scelta una legge di parete, ovvero una descrizione matematica del comportamento del fluido in prossimità della parete solida.

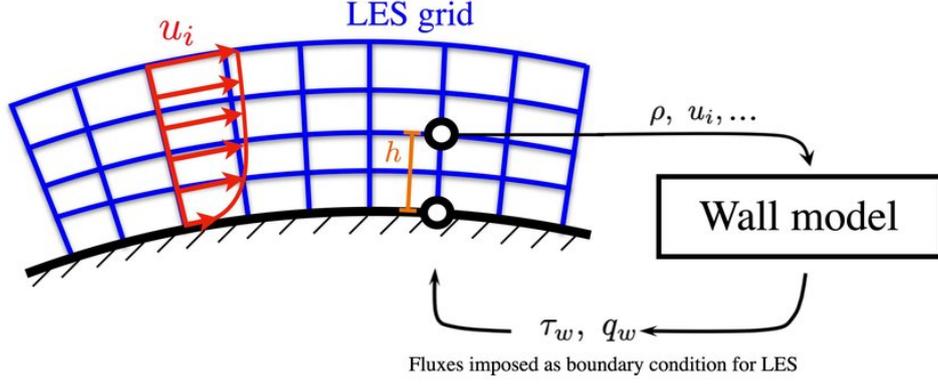


Figura 4.5: Schema del funzionamento del modello WMLES.

In questo lavoro di tesi viene scelta la legge di Reichard, utilizzata solitamente in canali turbolenti:

$$\frac{u}{u_\tau} = \frac{1}{k} \log(1 + ky^+) + 7.8 \left(1 - e^{-\frac{y^+}{11}} - \frac{y^+}{11} e^{-0.33y^+} \right) \quad (4.25)$$

ove u_τ è la velocità di attrito, y^+ è la distanza in unità viscosa dalla parete, quantità adimensionale, $k = 0.41$ è la costante di von Kármán.

Tramite un processo iterativo si ricava la velocità di attrito u_τ e, di conseguenza, la viscosità aggiuntiva:

$$\nu_a = \frac{\tau_w}{\frac{\partial u}{\partial z}} = \rho \frac{u_\tau^2}{\frac{\partial u}{\partial z}} \quad (4.26)$$

Infine, il modello WALE (*Wall-Adapting-Local-Eddy viscosity*) [20] viene utilizzato come modello di chiusura del modello WMLES in sostituzione al modello di Smagorinsky per il suo funzionamento più corretto in presenza delle pareti. Anche in questo caso il tensore degli sforzi viscosi viene rappresentato da una viscosità residua data da:

$$\nu_r = (C_w \Delta)^2 \frac{(S_{ij}^d S_{ij}^d)^{3/2}}{(\bar{S}_{ij} \bar{S}_{ij})^{5/2} + (S_{ij}^d S_{ij}^d)^{5/4}} \quad (4.27)$$

ove Δ è il *grid spacing*, C_w è una costante, mentre \bar{S}_{ij} è la parte simmetrica del gradiente di velocità e S_{ij}^d è la parte simmetrica senza traccia del quadrato del tensore del gradiente di velocità.

Capitolo 5

Metodo Numerico

In questo capitolo vengono presentati gli elementi che compongono il metodo numerico sviluppato. Alla base del codice vi è il solutore *CaNS* (*Canonical Navier-Stokes*) sviluppato da Pedro Costa nel 2018 [4], un metodo numerico per simulazioni rapide e parallelizzabili di flussi turbolenti: a questo vengono implementate la tecnica IBM, il modello WMLES e, soprattutto, il SDF. In Figura 5.1 viene mostrato in sintesi il funzionamento del codice con i vari elementi presenti.

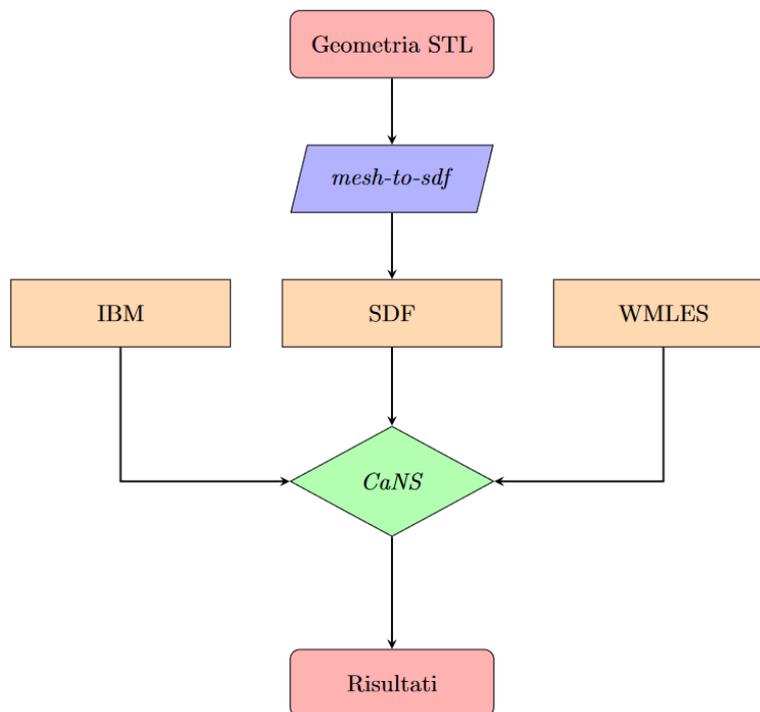


Figura 5.1: Schema del metodo numerico sviluppato.

5.1 Solutore *CaNS*

Il solutore *CaNS* risolve le equazioni di Navier-Stokes incomprimibili su una griglia Cartesiana strutturata ed equispaziata, con densità unitaria e viscosità cinematica ν :

$$\nabla \cdot \vec{u} = 0 \quad (5.1)$$

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\nabla p + \nu \nabla^2 \vec{u} \quad (5.2)$$

Per la discretizzazione spaziale viene implementato un metodo alle differenze finite al secondo ordine con una disposizione *staggered* dei punti della griglia, ovvero la *Marker and Cell* vista nel paragrafo 2.1. Le equazioni vengono quindi accoppiate tramite un metodo *pressure-correction* e integrate nel tempo tramite uno schema Runge-Kutta a 3 step (RK3¹).

Questo algoritmo prevede che, partendo dai campi di velocità e pressione noti al passo temporale n , venga calcolata una velocità predetta (*prediction velocity*) \vec{u}^* . Utilizzando lo schema temporale RK3 bisogna considerare che l'integrazione temporale avvenga in tre *substep*, ovvero $k = 1, 2, 3$, e che a $k = 1$ corrisponda il passo temporale $n = 1$ mentre a $k = 3$ quello $n + 1$:

$$u^* = u^k + \Delta t \left(\alpha_k \vec{A} \vec{D}^k + \beta_k \vec{A} \vec{D}^{k-1} - \gamma_k \nabla p^{k-1/2} \right) \quad (5.3)$$

ove $\vec{A} \vec{D} = -(\vec{u} \cdot \nabla) \vec{u} + \nu \nabla^2 \vec{u}$, mentre i coefficienti dello schema RK3 sono dati da $\alpha_k = \left\{ \frac{8}{15}, \frac{5}{12}, \frac{3}{4} \right\}$, $\beta_k = \left\{ 0, -\frac{17}{60}, -\frac{5}{12} \right\}$, $\gamma_k = \alpha_k + \beta_k$.

Successivamente viene risolta l'equazione di Poisson per la pressione, utilizzando la velocità predetta:

$$\nabla^2 \Phi = \frac{\nabla \cdot \vec{u}^*}{\gamma_k \Delta t} \quad (5.4)$$

ove Φ è la pressione di correzione (*pressure correction*).

Infine viene corretto il campo di velocità,

$$\vec{u}^k = \vec{u}^* - \gamma_k \Delta t \nabla \Phi \quad (5.5)$$

e il campo della pressione,

$$p^{k+1/2} = p^{k-1/2} + \Phi \quad (5.6)$$

¹Schema numerico alle differenze finite.

Al fine di ottenere una integrazione stabile si utilizza il seguente criterio sul *time step*:

$$\Delta t < \min \left(\frac{1.65\Delta r^2}{\nu}, \frac{\sqrt{3}\Delta r}{\max_{i,j,k}(|u| + |v| + |w|)} \right) \quad (5.7)$$

ove $\Delta r = \min(\Delta x, \Delta y, \Delta z)$, ovvero la minima spaziatura della griglia nelle tre direzioni.

5.1.1 Implementazione e Message-Passing Interface (MPI)

L'algoritmo numerico è implementato in FORTRAN90/95 con una estensione *Message-Passing Interface* (MPI) per la parallelizzazione dei processi. In generale MPI è un modello di programmazione e una libreria di standard per la comunicazione tra processi paralleli in calcolo ad alte prestazioni e applicazioni distribuite: Tramite una serie di comandi e funzioni standard di MPI si consente ai processi in esecuzione su nodi diversi di comunicare tra loro scambiandosi messaggi, questi messaggi possono includere dati, comandi o informazioni di stato.

In *CaNS* il dominio computazionale viene suddiviso in una serie di sottodomini attraverso una decomposizione *pencil-like*, ovvero si ottengono dei sottodomini cubici che rappresentano una sezione della griglia lungo una dimensione. In Figura 5.2 sono rappresentati alcuni esempi di decomposizione. Il metodo *pressure-correction* viene modificato per risolvere le equazioni tenendo in considerazione la decomposizione del dominio. Il procedimento è descritto più nel dettaglio in [4].

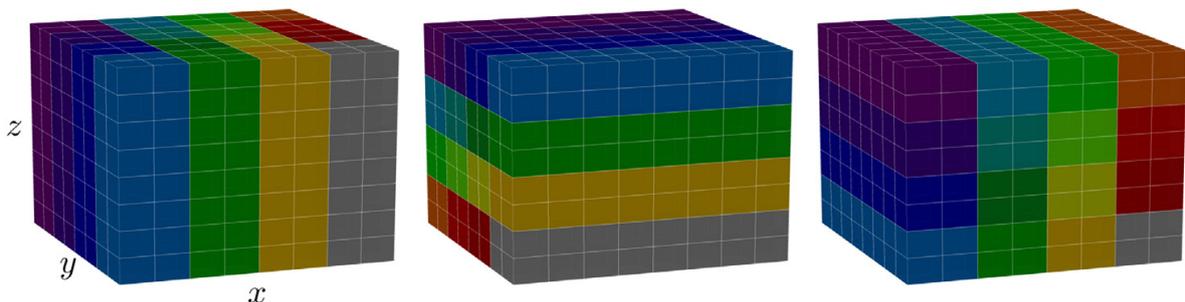


Figura 5.2: Esempi di decomposizione *pencil-like* [4].

5.2 Immersed Boundary Method

La tecnica IBM implementata nel codice sviluppato si basa sul metodo proposto da Breugem [2]. Questo metodo è stato derivato da una modifica della tecnica IBM di Uhlmann [28], che rientra nella categoria dei metodi *discrete forcing approach* (Vedi paragrafo 2.2.2). La tecnica implementa il seguente schema *pressure-correction* basato sullo schema Crank-Nicolson² per l'integrazione nel tempo:

$$\mathbf{u}^* = \mathbf{u}^n + \frac{\Delta t}{\rho_f} \left(-\nabla p^{n-1/2} + \mathbf{rhs}^{n+1/2} \right), \quad (5.8)$$

$$\mathbf{u}^{**} = \mathbf{u}^* + \Delta t \mathbf{f}^{n+1/2}, \quad (5.9)$$

$$\nabla^2 \tilde{p} = \frac{\rho_f}{\Delta t} \nabla \cdot \mathbf{u}^{**}, \quad (5.10)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \frac{\Delta t}{\rho_f} \nabla \tilde{p}, \quad (5.11)$$

$$p^{n+1/2} = p^{n-1/2} + \tilde{p}, \quad (5.12)$$

ove $\mathbf{rhs} \equiv -\nabla p_e - \rho_f \nabla \cdot \mathbf{u}\mathbf{u} + \mu_f \nabla^2 \mathbf{u}$, mentre \mathbf{u}^* e \mathbf{u}^{**} rappresentano la prima e la seconda *prediction velocity*, \tilde{p} è la *pressure correction* e Δt è il *time step* computazionale.

La *forcing function* viene calcolata tramite tre step:

1. Interpolazione della prima *prediction velocity* \mathbf{u}^* dalla griglia Euleriana alla griglia Lagrangiana (La prima appartiene all'IB, la seconda rappresenta il dominio computazionale, ci si fa riferimento al paragrafo 2.1 e viene rappresentata in Figura 5.3)
2. Calcolo della forza IBM sulla griglia Lagrangiana basato sulla differenza tra la prima *prediction velocity* interpolata e la velocità della particella.
3. Distribuzione di questa forza dalla griglia Lagrangiana alla griglia Euleriana.

Le operazioni di interpolazione e distribuzione si basano su una funzione delta di Dirac regolarizzata D , in questo caso questa viene estesa lungo tre celle della griglia in ognuna

²Anche chiamato metodo dei trapezi, è uno schema numerico alle differenze finite accurato al secondo ordine.

delle tre direzioni. Il calcolo si può riassumere con il seguente schema:

$$\mathbf{U}_l^* = \sum_{ijk} \mathbf{u}_{ijk}^* D(\mathbf{x}_{ijk} - \mathbf{X}_l^n) \Delta x \Delta y \Delta z, \quad (5.13)$$

$$\mathbf{F}_l^{n+1/2} = \frac{\mathbf{U}_p(\mathbf{X}_l^n) - \mathbf{U}_l^*}{\Delta t}, \quad (5.14)$$

$$\mathbf{f}_{ijk}^{n+1/2} = \sum_l \mathbf{F}_l^{n+1/2} \delta_d(\mathbf{x}_{ijk} - \mathbf{X}_l^n) \Delta V_l. \quad (5.15)$$

ove le quantità in maiuscolo sono calcolate sulla griglia Lagrangiana mentre quelle in minuscolo sulla griglia Euleriana. \mathbf{x}_{ijk} denota la posizione del punto della griglia Euleriana di indice (i, j, k) , \mathbf{X}_l denota la posizione del punto della griglia Lagrangiana con indice l . Δx , Δy e Δz sono le dimensioni della cella della griglia Euleriana e ΔV_l è il volume delle celle della griglia Lagrangiana.

La funzione delta di Dirac regolarizzata D scelta garantisce che la forza totale e il momento torcente che il fluido e le particelle esercitano l'uno sull'altro siano preservati nelle operazioni di interpolazione e distribuzione; tuttavia ciò vale solo quando in ciascuna direzione coordinata la dimensione corrispondente delle celle della griglia Euleriana è spazialmente uniforme. Di conseguenza viene scelta una griglia cartesiana uniforme per cui $\Delta x = \Delta y = \Delta z$.

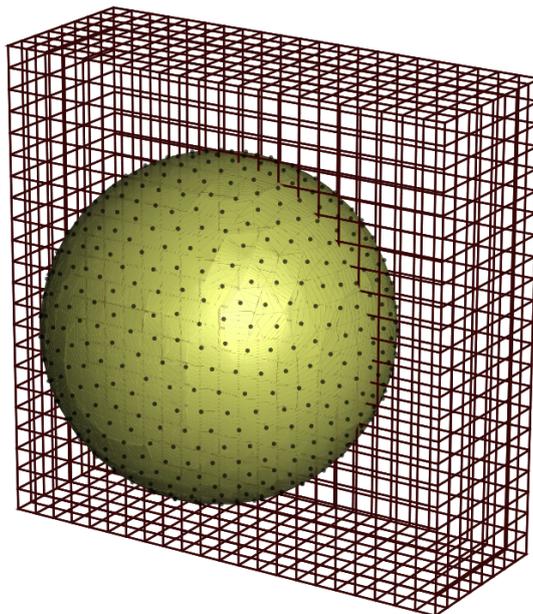


Figura 5.3: Rappresentazione della griglia Euleriana e Lagrangiana [2].

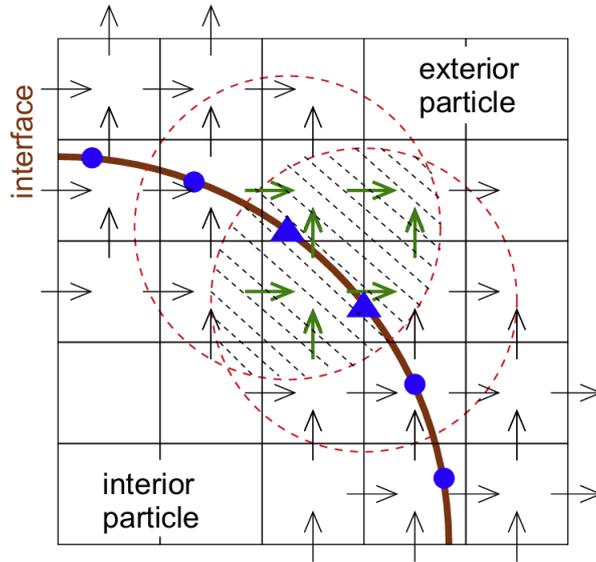


Figura 5.4: Sovrapposizione del range di azione di due punti Lagrangiani adiacenti [2].

L'implementazione della funzione D , delta di Dirac generalizzata, porta a due problematiche principali e per questa ragione Breugem [2] propone dei miglioramenti al metodo originale. La prima problematica riguarda il fatto che le operazioni di interpolazione e diffusione si traducono in una distribuzione diffusa della forza IBM attorno all'interfaccia di una particella. Questo fenomeno è illustrato in Figura 5.4. Le frecce indicano tutti i punti della griglia Euleriana che sono coinvolti nella forzatura della velocità nei punti della griglia Lagrangiana sull'interfaccia. I due cerchi mostrano il raggio d'azione della

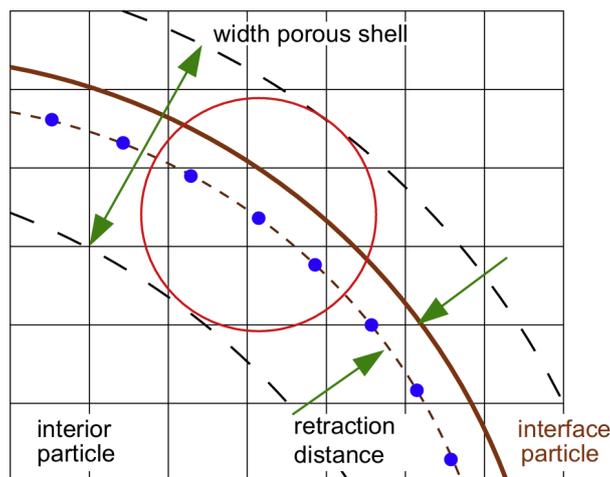


Figura 5.5: Raffigurazione della distanza di retrazione applicata ai punti della griglia Lagrangiana [2].

funzione delta per i due punti della griglia Lagrangiana indicati dai triangoli. I cerchi si sovrappongono l'uno all'altro, il che significa che i punti della griglia Euleriana nella regione di sovrapposizione vengono utilizzati per forzare la velocità in entrambi i punti della griglia Lagrangiana. In questi punti della griglia Euleriana la forzatura richiesta per la velocità delle particelle desiderata in uno dei punti della griglia Lagrangiana è perturbata dalla forzante necessaria per l'altro e viceversa. Come conseguenza di questa sovrapposizione nella forzatura, la distribuzione della forza IBM attorno all'interfaccia della particella potrebbe non forzare molto bene la velocità desiderata della particella.

Per evitare questo problema viene utilizzato uno schema iterativo, chiamato *multidirect forcing*, in cui la forza IBM viene trovata attraverso N_s iterazioni. Nel caso specifico della tesi specifico sono state utilizzate 2 iterazioni, per non aumentare in modo significativo il costo computazionale.

La seconda problematica derivante dall'uso della funzione D è il fatto che l'interfaccia tra flusso e oggetto non risulta una superficie continua ma un guscio poroso che, nel caso specifico, ha una larghezza di tre celle della griglia. Questo guscio poroso ha due effetti principali: da un lato aumenta la dimensione delle geometrie e quindi aumenta la forza di *Drag*; dall'altro la porosità della parete in generale tende a far diminuire il *Drag*. Per risolvere questo problema la griglia Lagrangiana viene leggermente retratta dalla superficie verso l'interno della particella. Questo viene illustrato in Figura 5.5. Tramite dei calcoli sulle forze di *Drag* e alcune simulazioni numeriche viene ottenuto il valore ottimale della distanza di retrazione.

Il metodo numerico finale viene ottenuto tramite un'integrazione del tempo con un Runge-Kutta a tre stadi per tutti i termini delle equazioni di Navier-Stokes eccetto il gradiente di pressione per il quale è stato utilizzato lo schema Crank-Nicolson; inoltre vengono considerate le due principali modifiche rispetto al metodo originale. Lo schema *pressure-correction* risulta ora:

do $q = 1, 3$

$$\mathbf{u}^* = \mathbf{u}^{q-1} + \frac{\Delta t}{\rho_f} \left(-(\alpha_q + \beta_q) \nabla p^{q-3/2} + \alpha_q \mathbf{RHS}^{q-1} + \beta_q \mathbf{RHS}^{q-2} \right),$$

$$\mathbf{U}_1^* = \sum_{ijk} \mathbf{u}_{ijk}^* \delta_d (\mathbf{x}_{ijk} - \mathbf{X}_l^{q-1}) \Delta x \Delta y \Delta z,$$

$$\mathbf{F}_l^{q-1/2,0} = \frac{\mathbf{U}_p(\mathbf{X}_l^{q-1}) - \mathbf{U}_l^*}{\Delta t},$$

$$\mathbf{f}_{ijk}^{q-1/2,0} = \sum_l \mathbf{F}_l^{q-1/2,0} \delta_d (\mathbf{x}_{ijk} - \mathbf{X}_l^{q-1}) \Delta V_l$$

do $s = 1, N_s$

$$\mathbf{U}_l^{**,s-1} = \sum_{ijk} \mathbf{u}_{ijk}^{**,s-1} \delta_d (\mathbf{x}_{ijk} - \mathbf{X}_i^{q-1}) \Delta x \Delta y \Delta z,$$

$$\mathbf{F}_l^{q-1/2,s} = \mathbf{F}_l^{q-1/2,s-1} + \frac{\mathbf{U}_p(\mathbf{X}_l^{q-1}) - \mathbf{U}_i^{**,s-1}}{\Delta t},$$

$$\mathbf{f}_{ijk,l}^{q-1/2,s} = \sum_l \mathbf{F}_l^{q-1/2,s} \delta_d (\mathbf{x}_{ijk} - \mathbf{X}_l^{q-1}) \Delta V_l,$$

$$\mathbf{u}^{**,s} = \mathbf{u}^* + \Delta t \mathbf{f}_{ijk}^{q-1/2,s},$$

enddo

$$\nabla^2 \tilde{p} = \frac{\rho_f}{(\alpha_q + \beta_q) \Delta t} \nabla \cdot \mathbf{u}^{**,N_s},$$

$$\mathbf{u}^q = \mathbf{u}^{**,N_s} - \frac{(\alpha_q + \beta_q) \Delta t}{\rho_f} \nabla \tilde{p},$$

$$p^{q-1/2} = p^{q-3/2} + \tilde{p},$$

enddo

Come per lo schema RK3 implementato in *CaNS*, anche in questo caso lo step Renge-Kutta q corrisponde al *time step* n per $q = 0$ e a $n + 1$ per $q = 3$; $\mathbf{u}^{**,s}$ rappresenta la seconda *prediction velocity* allo step s di iterazione dello schema di *multidirect forcing*. I coefficienti di Runge-Kutta sono gli stessi riportati nel paragrafo 5.1.

Le equazioni sono quindi discretizzate nello spazio con un metodo ai volumi finiti: Lo schema risulta accurato almeno al secondo ordine per il calcolo di velocità e pressione sia nello spazio che nel tempo.

5.3 Signed Distance Field

Il *Signed Distance Field* viene prima calcolato attraverso un codice sviluppato in linguaggio PYTHON e poi implementato in FORTRAN come *subroutine* del codice base *CaNS*.

Il codice FORTRAN richiede come input la geometria in formato STL e le dimensioni del dominio computazionale, ovvero le lunghezze $L_x \times L_y \times L_z$ e il numero di punti $n_x \times n_y \times n_z$. Attraverso il modulo *mesh-to-sdf* (Paragrafo 3.3.1) viene creato il SDF per ogni punto del dominio. Il SDF in output viene ottenuto in due formati: Un file in formato BIN che diventa un input della subroutine di *CaNS* e l'altro in formato VTR che può essere utilizzato per visualizzare il campo tramite il software *Paraview*.

L'implementazione in FORTRAN si sviluppa in due passaggi principali:

1. Il SDF viene *scansionato* lungo ognuna delle tre direzioni al fine di trovare i punti del dominio per cui il campo ha valore nullo.
2. Per ognuno dei punti trovati viene calcolata la normale uscente dalla superficie utilizzando il gradiente di SDF (Proprietà descritta nel paragrafo 3.2).

Il risultato è una nuvola di punti, con relative normali, che rappresentano l'IB da utilizzare nella simulazione: Queste vengono utilizzate come *input* per la tecnica IBM e la WMLES.

Tra i vantaggi nell'utilizzo del SDF elencati nell'introduzione al Capitolo 1 è presente la possibilità di rappresentare il SDF sulla griglia computazionale con una risoluzione arbitraria indipendentemente dalla geometria STL utilizzata. Per illustrare meglio questo vantaggio è stato calcolato il SDF di una geometria imponendo quattro diverse risoluzioni Δ della griglia in Figura 5.6

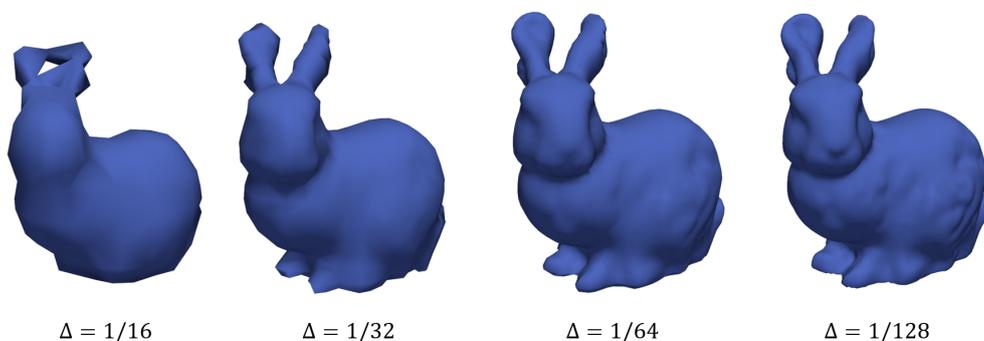


Figura 5.6: SDF calcolato al variare della risoluzione.

Capitolo 6

Validazione Codice

Nel presente capitolo il codice sviluppato viene testato attraverso una serie di casi di riferimento per esplorarne e comprenderne il comportamento rispetto ai risultati disponibili in letteratura. In particolare vengono studiati due casi: il flusso su cilindro, il quale può essere considerato come un caso bidimensionale per la geometria a cui si fa riferimento, e il flusso su una sfera, al fine di testare una implementazione tridimensionale del codice; la vasta disponibilità di articoli scientifici e studi per ciascuno di questi casi consente di ottenere una validazione accurata del codice. Per ottenere un confronto tra i dati ottenuti tramite simulazione e la letteratura vengono calcolati il coefficiente di Drag C_D al variare del numero di Reynolds Re e, in determinate situazioni, le dimensioni della zona di ricircolo generata dalla geometria. Tutte le simulazioni vengono condotte su un canale di dimensioni $L_x \times L_y \times L_z$ e con un determinato numero di celle. Sulle due facce del

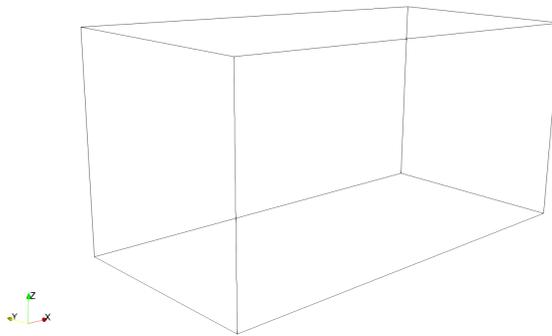


Figura 6.1: Dominio computazionale.

dominio parallele al piano xz sono state imposte delle condizioni al contorno periodiche, mentre nelle altre quattro facce condizioni al contorno di Dirichlet o Neumann a seconda dei casi, tutto questo per ognuna delle tre componenti della velocità u , v e w e la pressione p . La velocità di ingresso unitaria è stata imposta in modo tale che il flusso sia parallelo all'asse x e si sviluppi per valori crescenti dell'ascissa.

6.1 Flusso su Cilindro

Il flusso su cilindro è un utile punto di riferimento poiché la dinamica del flusso è governata dalla separazione dello strato limite dalla superficie del cilindro e quindi una rappresentazione errata dell'IB porterebbe ad una alterazione della dinamica.

Inizialmente viene scelta la geometria del cilindro: il diametro viene imposto a $d = 0.2$ mentre l'altezza a $h = 1$, successivamente viene eseguito il *meshing* della superficie con un numero di elementi pari a circa 15000 al fine di ottenere il file STL per la simulazione (Figura 6.2).

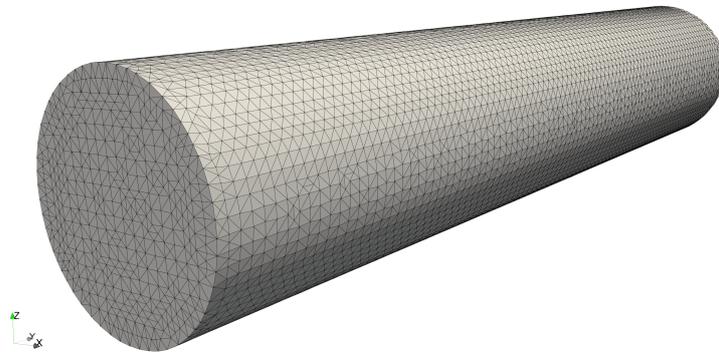
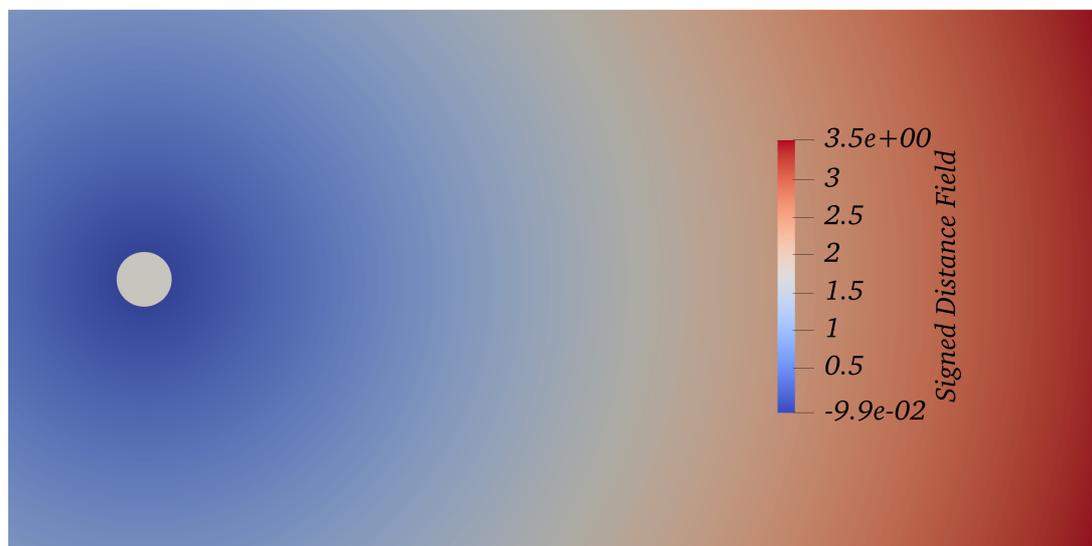


Figura 6.2: Geometria STL del cilindro.

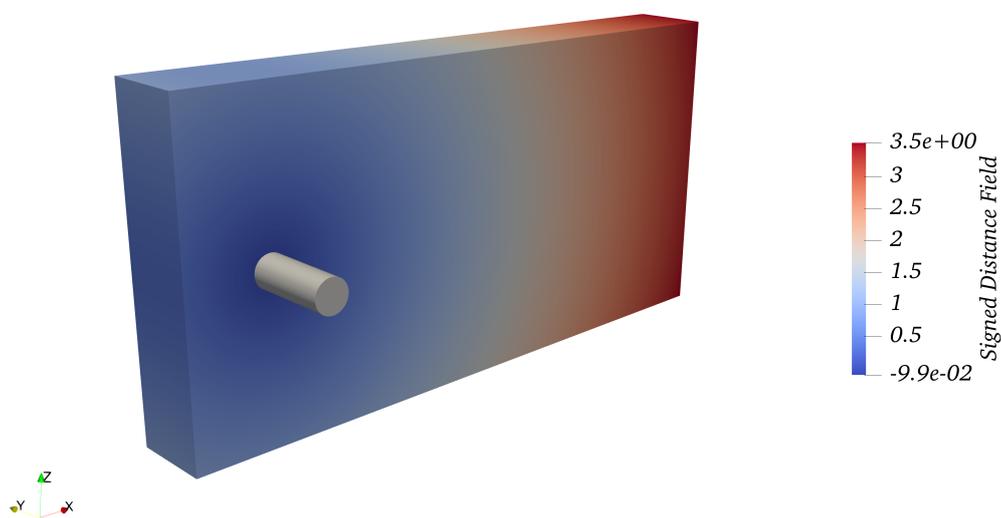
6.1.1 Calcolo SDF

Il cilindro viene posto in un grande dominio computazionale di dimensioni $L_x \times L_y \times L_z = 4 \times 1 \times 2$ composto da una griglia regolare di $n_x \times n_y \times n_z = 512 \times 128 \times 256$ punti con lo scopo di ottenere una rappresentazione accurata sia della zona di ricircolo per bassi numeri di Reynolds, sia della scia di von Kármán per alti numeri di Reynolds.

Il *signed distance field* viene generato a partire da questo file STL e ciò che si ottiene viene riportato nelle Figure 6.3a e 6.3b: La distribuzione dei valori nel dominio è radialmente simmetrica e crescente, e ciò dimostra che la geometria che ha generato il SDF è cilindrica.



(a) SDF ottenuto dalla geometria cilindrica.



(b) Sezione di SDF.

Figura 6.3: SDF del cilindro.

6.1.2 Simulazione e Risultati

Il numero di Reynolds per un flusso che investe un cilindro vale:

$$Re = \frac{v d}{\nu} \quad (6.1)$$

ove $d = 0.2$ è il diametro del cilindro, $v = 1$ è la velocità di ingresso, ν è la viscosità cinematica. In accordo con lo studio di riferimento la simulazione viene condotta per tre numeri di Reynolds differenti: 20, 40 e 200.

Nelle Figure 6.5-6.7 vengono riportati i grafici di velocità u , vorticità e le relative *streamlines*¹ per ognuno dei tre casi considerati. Per eseguire un confronto tra il modello sviluppato e i dati disponibili in letteratura viene calcolato il coefficiente di resistenza C_D , in questo caso dato da:

$$C_D = \frac{F_x}{\frac{1}{2} \rho v^2 d} \quad (6.2)$$

ove F_x è la forza in direzione x , risultato ottenibile dalla simulazione.

Inoltre, per $Re = 20$ e $Re = 40$, viene anche calcolata la dimensione della zona di ricircolo: Facendo riferimento alla Figura 6.4, i parametri l , a e b rappresentano rispettivamente la lunghezza della zona di ricircolo, la distanza dal cilindro al centro del vortice di scia e la distanza tra i centri dei vortici. L'angolo di separazione è indicato con θ ed è misurato a partire dall'asse x . I risultati vengono presentati in Tabella 6.1 facendo un confronto con tre studi differenti: Per i due casi a basso numero di Reynolds vi è un'ottima corrispondenza sia sulle dimensioni della zona di ricircolo, sia sul coefficiente di resistenza calcolato.

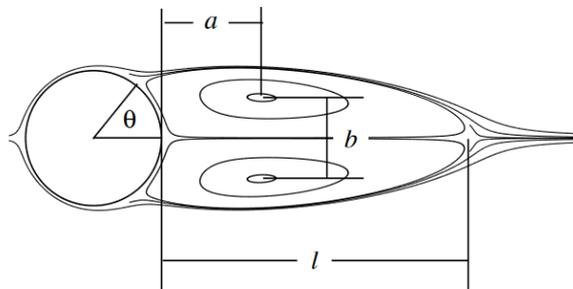
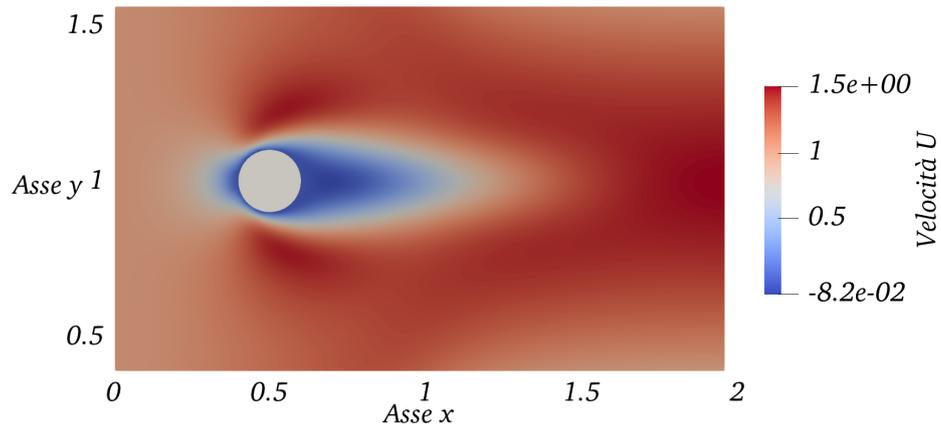
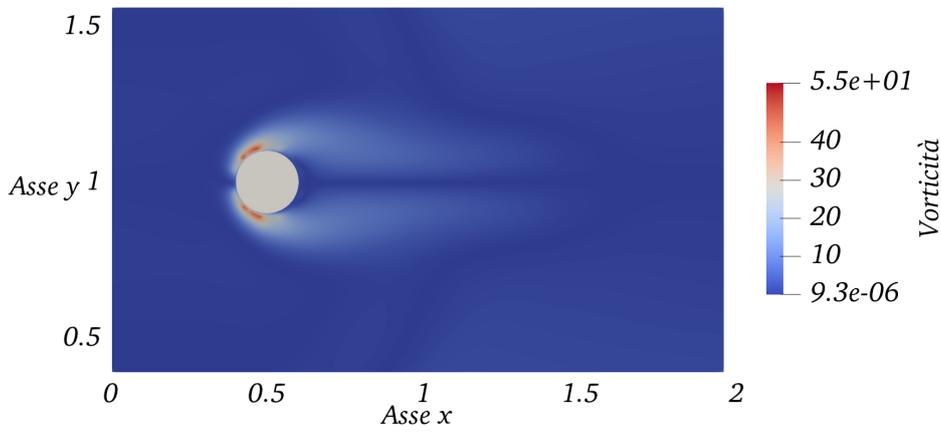


Figura 6.4: Parametri della zona di ricircolo.

¹Una *streamline*, in italiano linea di corrente, è il percorso tracciato da una particella che si muove con il flusso.



(a) Velocità $Re = 20$.

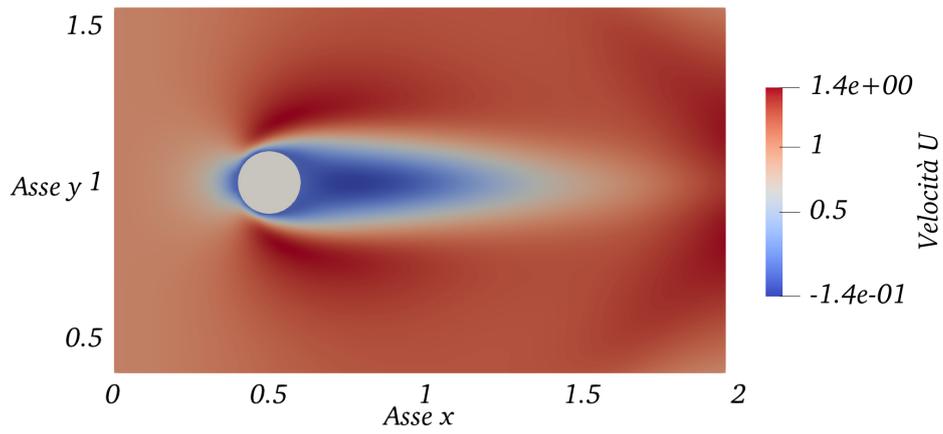


(b) Vorticità $Re = 20$.

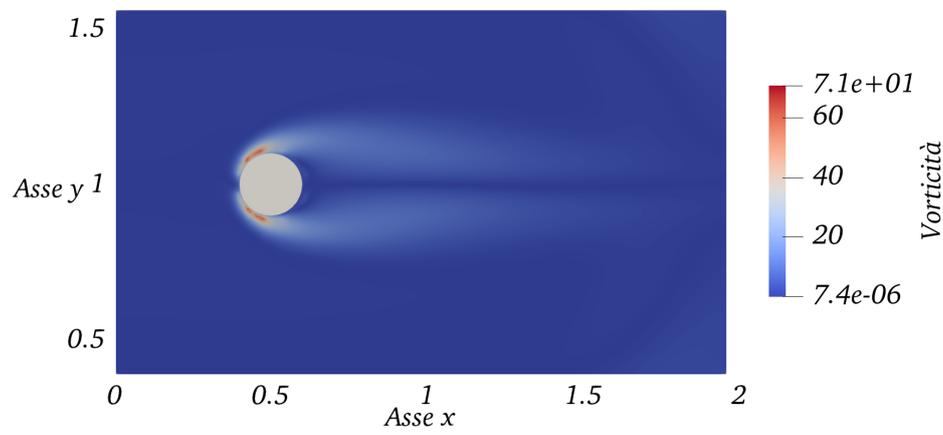


(c) Streamlines $Re = 20$.

Figura 6.5: Flusso su cilindro, risultati $Re = 20$.



(a) Velocità $Re = 40$.

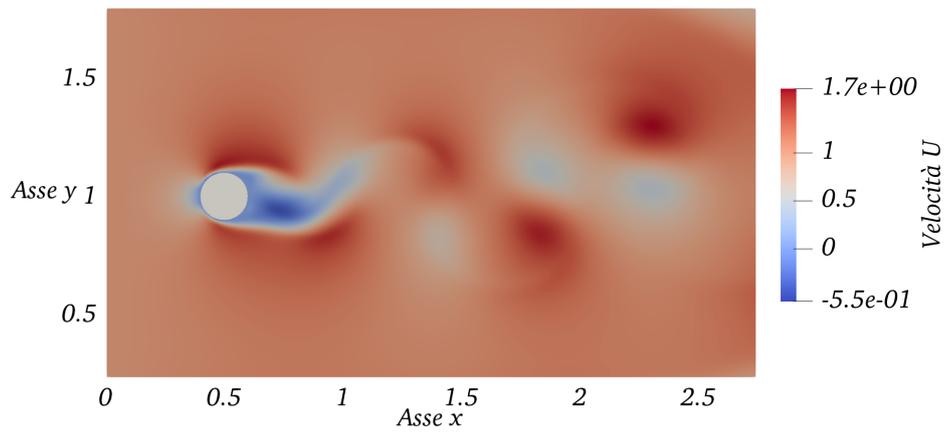


(b) Vorticità $Re = 40$.

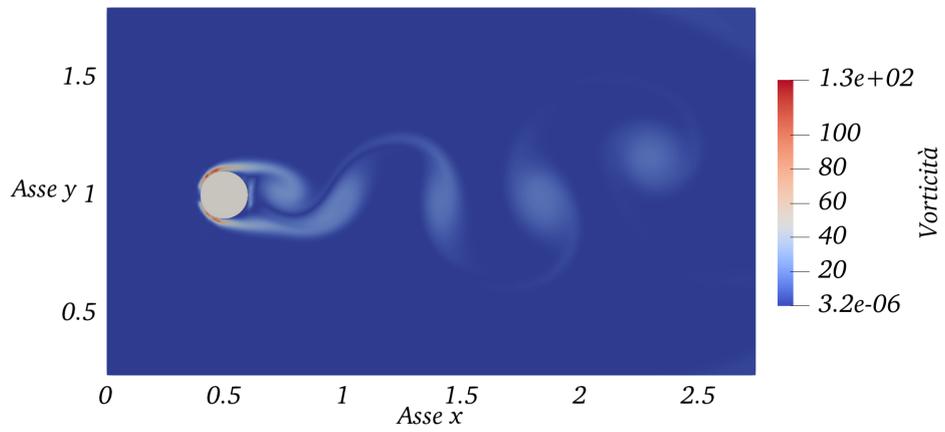


(c) *Streamlines* $Re = 40$.

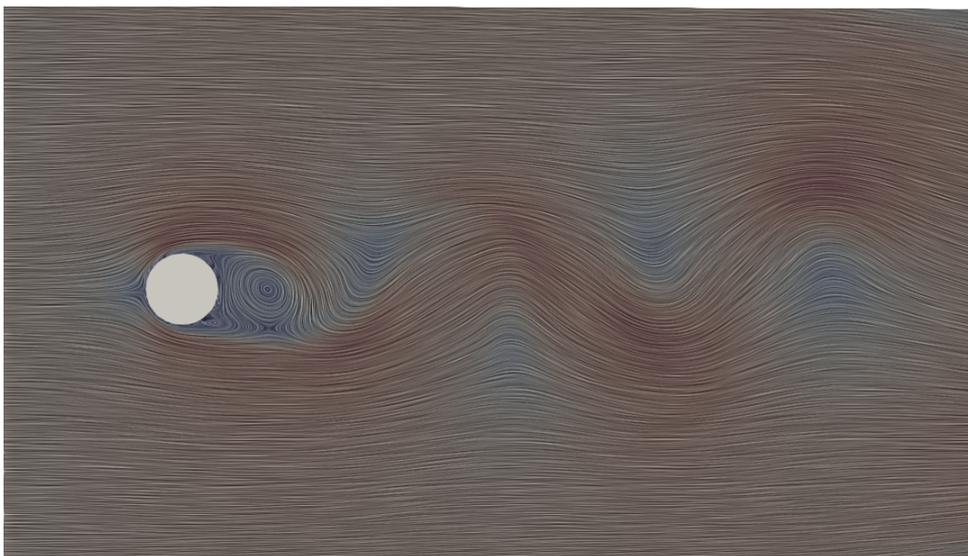
Figura 6.6: Flusso su cilindro, risultati $Re = 40$.



(a) Velocità $Re = 200$.



(b) Vorticità $Re = 200$.



(c) *Streamlines* $Re = 200$.

Figura 6.7: Flusso su cilindro, risultati $Re = 200$.

		l/d	a/d	b/d	α	C_D
$Re = 20$	Simulazione IBM+SDF	0.9	0.35	0.41	45.0°	2.03
	Taira and Colonius [27]	0.94	0.37	0.43	43.3°	2.06
	Linnick and Fasel [15]	0.93	0.36	0.43	43.5°	2.06
	Dennis and Chang [5]	0.94	–	–	43.7°	2.05
$Re = 40$	Simulazione IBM+SDF	2.12	0.75	0.56	50.0°	1.50
	Taira and Colonius [27]	2.30	0.73	0.60	53.7°	1.54
	Linnick and Fasel [15]	2.28	0.72	0.60	53.6°	1.54
	Dennis and Chang [5]	2.35	–	–	53.8°	1.52

Tabella 6.1: Confronto con letteratura della dimensione della zona di ricircolo e coefficiente di Drag per $Re = 20$ e $Re = 40$.

Infine, in Tabella 6.2 vengono presentati i risultati per il caso $Re = 200$. Poiché a questo numero di Reynolds il flusso su cilindro genera la scia di von Kármán, viene calcolato anche il numero di Strouhal definito da:

$$St = \frac{f_s d}{v} \quad (6.3)$$

ove f_s è la frequenza di distacco dei vortici nella scia di von Kármán. Inoltre viene calcolato anche il coefficiente di Drag, compreso tra due estremi.

		St	C_D
$Re = 200$	Simulazione IBM+SDF	0.21	1.29 ± 0.1
	Taira and Colonius [27]	0.196	1.35 ± 0.048
	Linnick and Fasel [15]	0.197	1.34 ± 0.044

Tabella 6.2: Confronto con letteratura del coefficiente di Drag per $Re = 200$.

6.2 Flusso su Sfera

Il flusso incompressibile su una sfera è un altro problema per il quale sono disponibili molti dati in letteratura, nel caso studiato ci si riferisce allo studio del 1999 di Johnson e Patel [9]. Questa tipologia di problema può essere considerato una estensione tridimensionale del flusso su cilindro e, in questo caso, le instabilità possono generare dei flussi instabili tridimensionali che hanno cinematica e interazioni vorticose molto complesse.

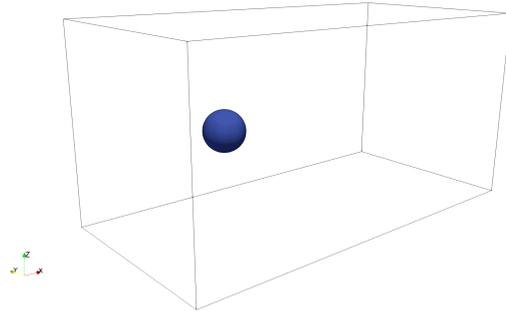
6.2.1 Calcolo SDF

Una sfera di diametro $d = 0.2$ (Figura 6.8) viene posta in un dominio di dimensioni $L_x \times L_y \times L_z = 2 \times 1 \times 1$ composto da una griglia regolare di $n_x \times n_y \times n_z = 256 \times 128 \times 128$. Il passo della griglia che si ottiene è di $\Delta s = 0.0078125$ e può essere considerato sufficiente per i risultati che si mira ad ottenere.

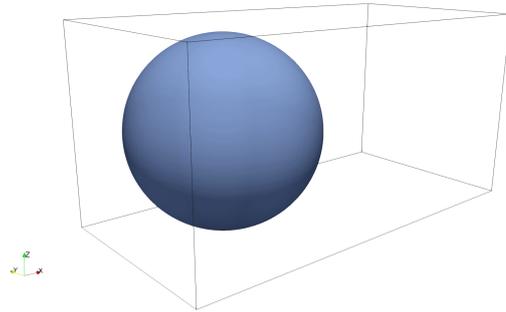
Utilizzando il modulo *mesh-to-sdf* viene calcolato il SDF della sfera all'interno del dominio preso in considerazione. Anche in questo caso la distribuzione dei valori è radialmente crescente, in accordo con i risultati attesi: per poter osservare meglio questa proprietà è stato eseguito un *contour plot* per alcuni valori di SDF a partire dal valore nullo (il SDF in questo caso è congruente alla superficie sferica) fino a 1.2 (Figura 6.9).



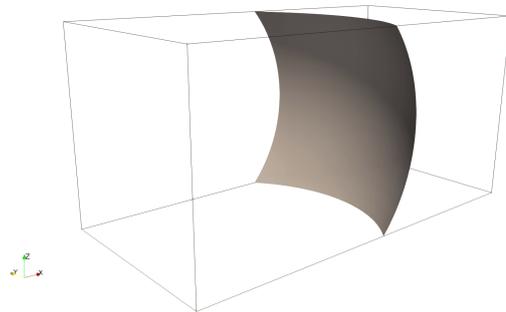
Figura 6.8: Geometria STL della sfera.



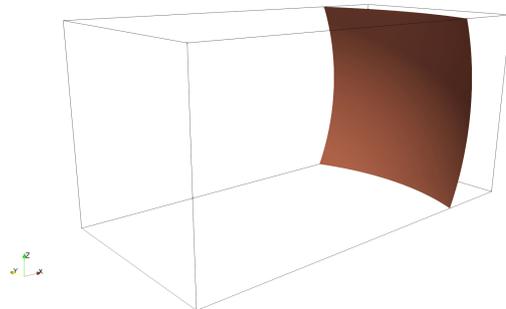
(a) $SDF = 0$



(b) $SDF = 0.35$



(c) $SDF = 0.8$



(d) $SDF = 1.2$

Figura 6.9: *Contour Plot* per quattro valori di SDF della sfera.

6.2.2 Simulazione e Risultati

Considerando che il flusso investe una superficie sferica, in questo caso il numero di Reynolds vale:

$$Re = \frac{v d}{\nu} \quad (6.4)$$

ove d è il diametro della sfera, $v = 1$ è la velocità di ingresso, ν è la viscosità cinematica. In accordo con lo studio di riferimento la simulazione viene condotta per 5 numeri di Reynolds diversi compresi tra 25 e 200.

In Figura 6.11 vengono rappresentate le *streamlines* per valori crescenti di Re , prendendo in considerazione la sezione mediana del dominio. Questa figura illustra chiaramente lo sviluppo della zona di ricircolo dietro la sfera. Si può osservare come, all'aumentare del numero di Reynolds, la lunghezza della separazione tenda ad aumentare. Tale fenomeno è particolarmente significativo poiché evidenzia l'influenza del numero di Reynolds sulla formazione e sull'estensione delle zone di ricircolo, fornendo una comprensione approfondita del comportamento del flusso intorno alla sfera.

Anche in questo caso si ottengono i valori di F_x come output della simulazione e viene calcolato il coefficiente di resistenza C_D , in questo caso dato da:

$$C_D = \frac{F_x}{\frac{1}{2} \rho v^2 \pi \frac{d^2}{4}} \quad (6.5)$$

e in Figura 6.10 viene riportato l'andamento di C_D rispetto a Re , confrontando i valori ottenuti dalla simulazione con quelli di riferimento. Il grafico evidenzia una buona corri-

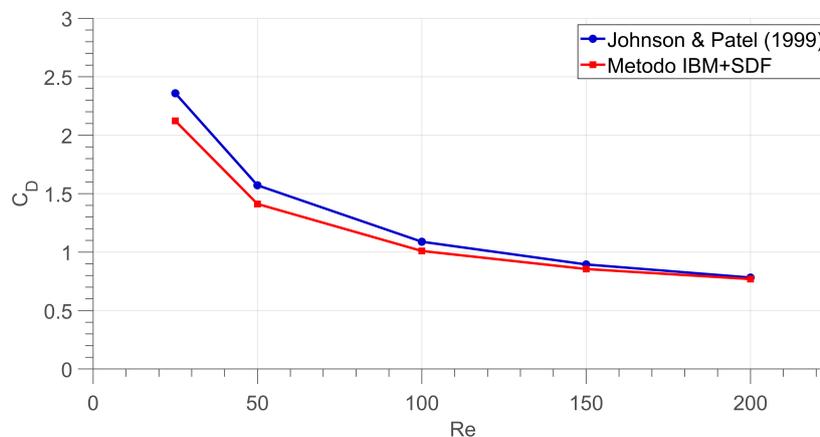


Figura 6.10: C_D vs Re per flusso su sfera.

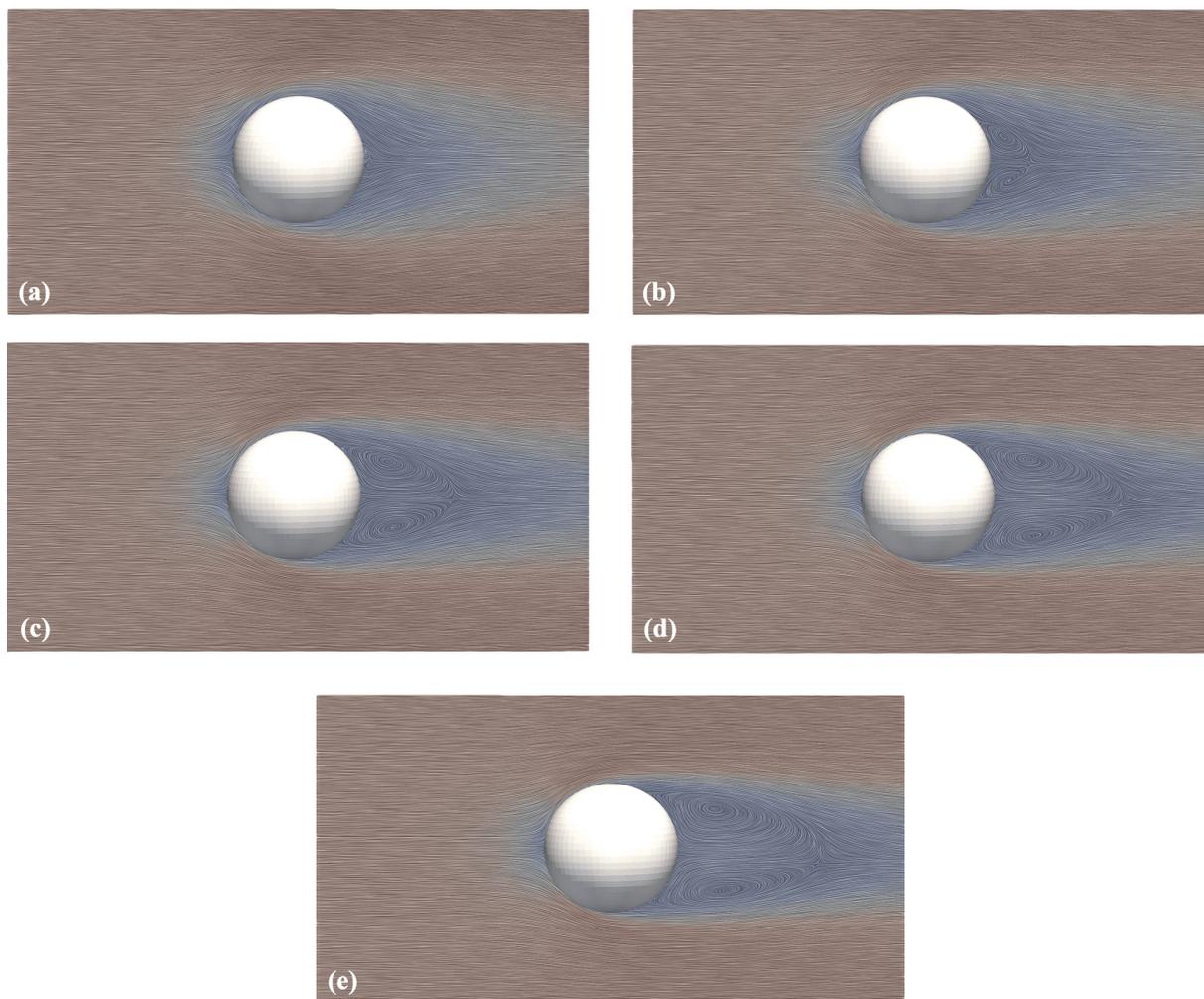


Figura 6.11: Rappresentazione delle *streamlines* per valori crescenti del numero di Reynolds: (a) $Re = 25$; (b) $Re = 50$; (c) $Re = 100$; (d) $Re = 150$; (e) $Re = 200$

spondenza tra i valori calcolati con il metodo IBM+SDF e i dati di riferimento, mostrando qualche discrepanza per i valori più bassi del numero di Reynolds.

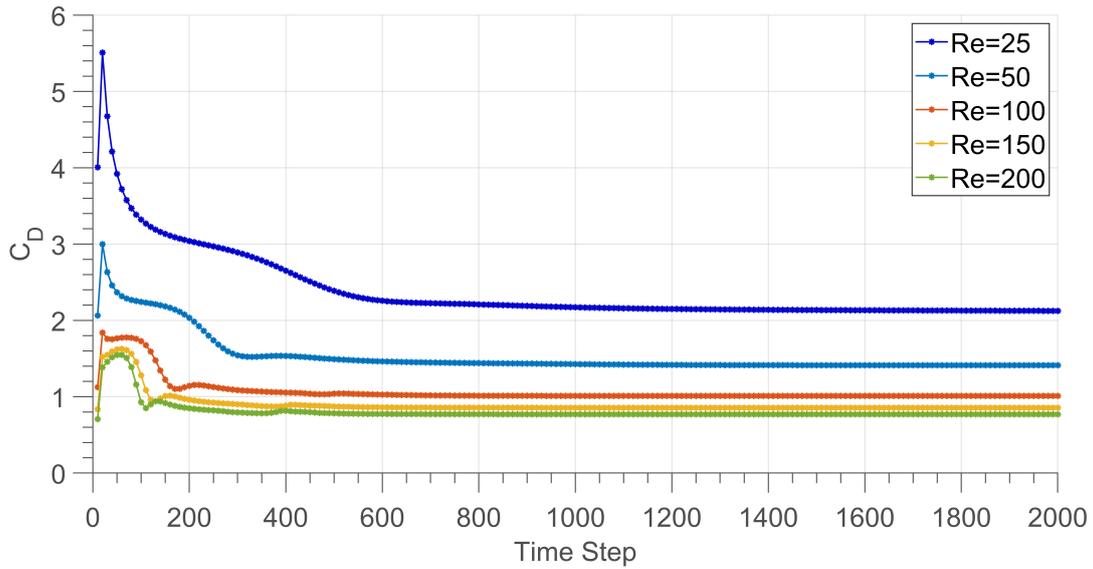


Figura 6.12: Andamenti C_D vs *timestep*.

Riferendosi nuovamente ai risultati ottenuti dalla simulazione, in Figura 6.12 vengono riportati gli andamenti di C_D rispetto al *time step* della simulazione provando che il metodo converge per ognuno dei cinque casi considerati. Inoltre per $Re = 100$ vengono confrontati i valori della lunghezza di separazione l e della posizione del centro del vortice della zona di ricircolo a e b con quelli di vari studi disponibili in letteratura (Tabella 6.3).

	l/d	a/d	b/d	C_D
Simulazione IBM+SDF	0.825	0.75	0.565	1.02
Johnson and Patel [9]	0.86	0.76	0.58	1.08
Marella et al. [17]	0.88	-	-	1.06

Tabella 6.3: Confronto con letteratura della dimensione della zona di ricircolo e coefficiente di Drag per $Re = 100$.

Capitolo 7

Risultati Qualitativi

Dopo aver completato la fase di validazione del codice, il focus di questo capitolo si sposta verso l'esplorazione qualitativa dei flussi turbolenti in geometrie complesse. L'obiettivo principale è quello di investigare come il codice sviluppato si comporta in contesti realistici, evidenziando le sue capacità di rappresentare fenomeni complessi come vortici, zone di separazione e interazioni complesse tra fluido e superfici solide.

Le analisi vengono condotte prima su una autovettura e poi in un ambiente urbano; il procedimento è lo stesso dei casi di validazione, descritto dettagliatamente nel capitolo 5. In entrambi i casi le geometrie vengono poste all'interno del dominio di dimensioni $L_x \times L_y \times L_z$ di Figura 6.1, tuttavia vengono imposte delle condizioni al contorno periodiche sulle due facce e viene considerato un flusso di Poiseuille all'ingresso; questo è un tipo specifico di flusso laminare, stazionario e viscoso, caratterizzato da un profilo di velocità parabolico lungo la sezione trasversale del dominio, con massima velocità al centro e velocità nulla in prossimità delle pareti. Infine, il numero di Reynolds viene imposto nel solutore *CaNS* come l'inverso della viscosità cinematica:

$$Re_{CaNS} = \frac{1}{\nu} \tag{7.1}$$

7.1 Flusso su Automobile

Seguendo lo stesso procedimento dei due casi di validazione del modello, viene considerata la geometria di una autovettura, dopo aver creato la *mesh* viene calcolato il SDF. In Figura 7.1 viene disegnato il *contour plot* per il valore di $SDF = 0$, ottenendo la geometria della automobile utilizzata come input nel solutore *CaNS*.

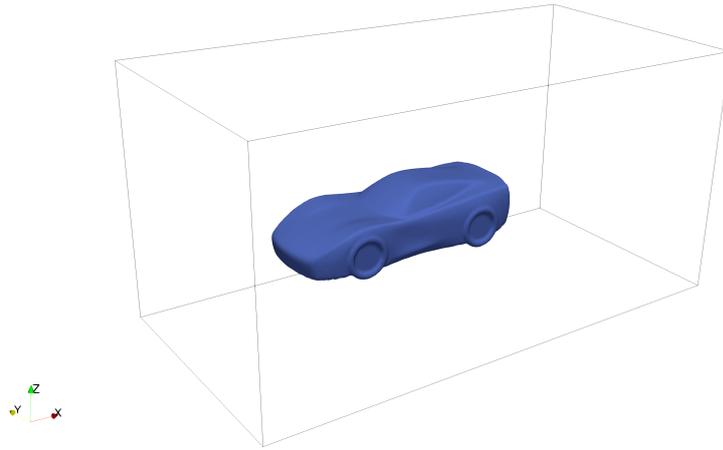
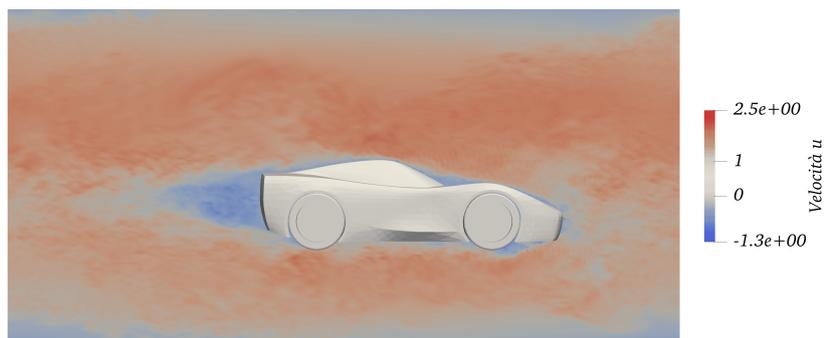


Figura 7.1: *Contout Plot* per il valore di $SDF = 0$.

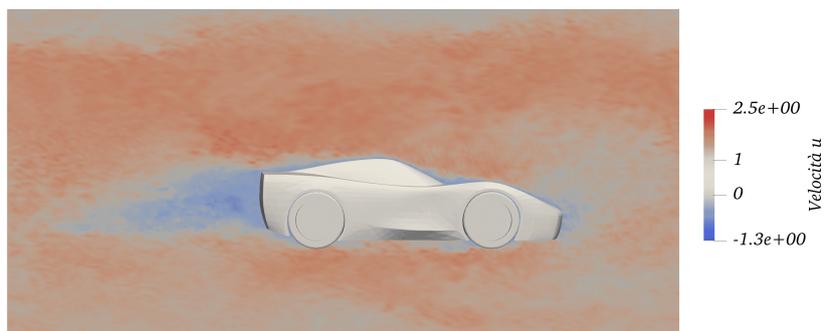
Il dominio è stato scelto di dimensioni $L_x \times L_y \times L_z = 2 \times 1 \times 1$ e con un numero di punti $n_x \times n_y \times n_z = 256 \times 128 \times 128$, mentre per il numero di Reynolds è stato scelto un valore di $Re = 10^5$.

In Figura 7.2 vengono riportati i profili di velocità u per quattro momenti differenti della simulazione: in questo caso vale $t_n = n * 1000$, ove n varia da 1 a 4 e t_n equivale alla iterata temporale della simulazione. Per gli stessi tempi vengono calcolate le *streamlines* in Figura 7.3.

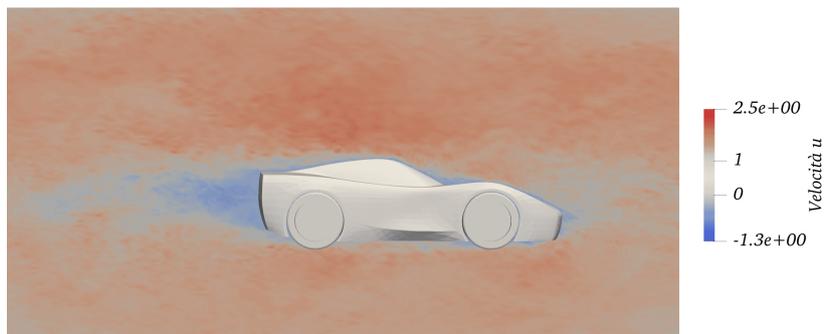
Inoltre, al fine di illustrare la complessità dell'esempio considerato, vengono riportate in Figura 7.4 le *streamlines* attorno all'autovettura, con una mappa di colori che rappresenta la velocità u . Viene posta maggiore attenzione alla zona di recircolo sviluppata dietro al veicolo, il flusso qui rallenta e si formano dei vortici.



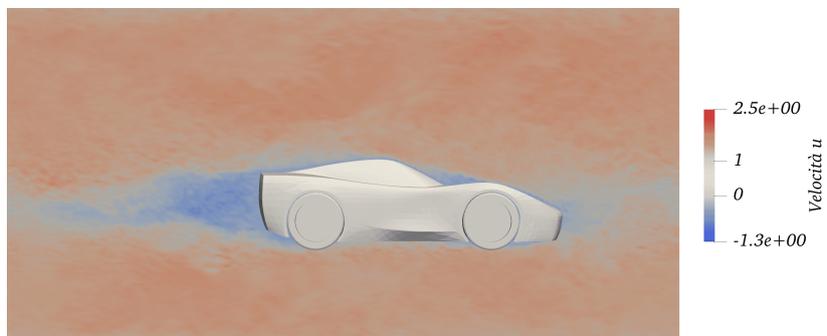
(a) $t = t_1$.



(b) $t = t_2$.



(c) $t = t_3$.

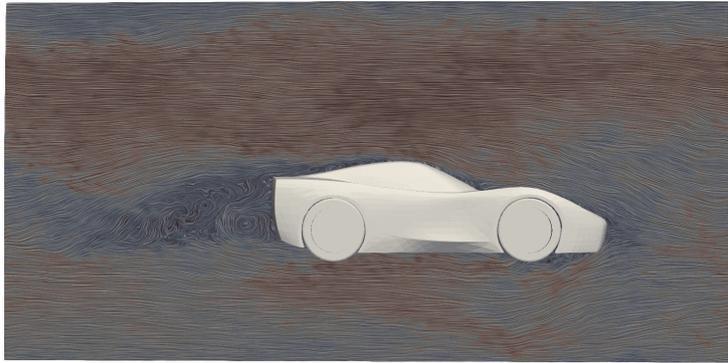


(d) $t = t_4$.

Figura 7.2: Velocità u a quattro tempi differenti.



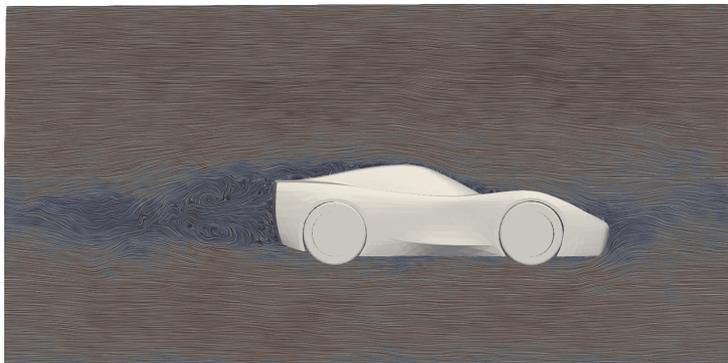
(a) $t = t_1$.



(b) $t = t_2$.



(c) $t = t_3$.



(d) $t = t_4$.

Figura 7.3: *Streamlines* a quattro tempi differenti.

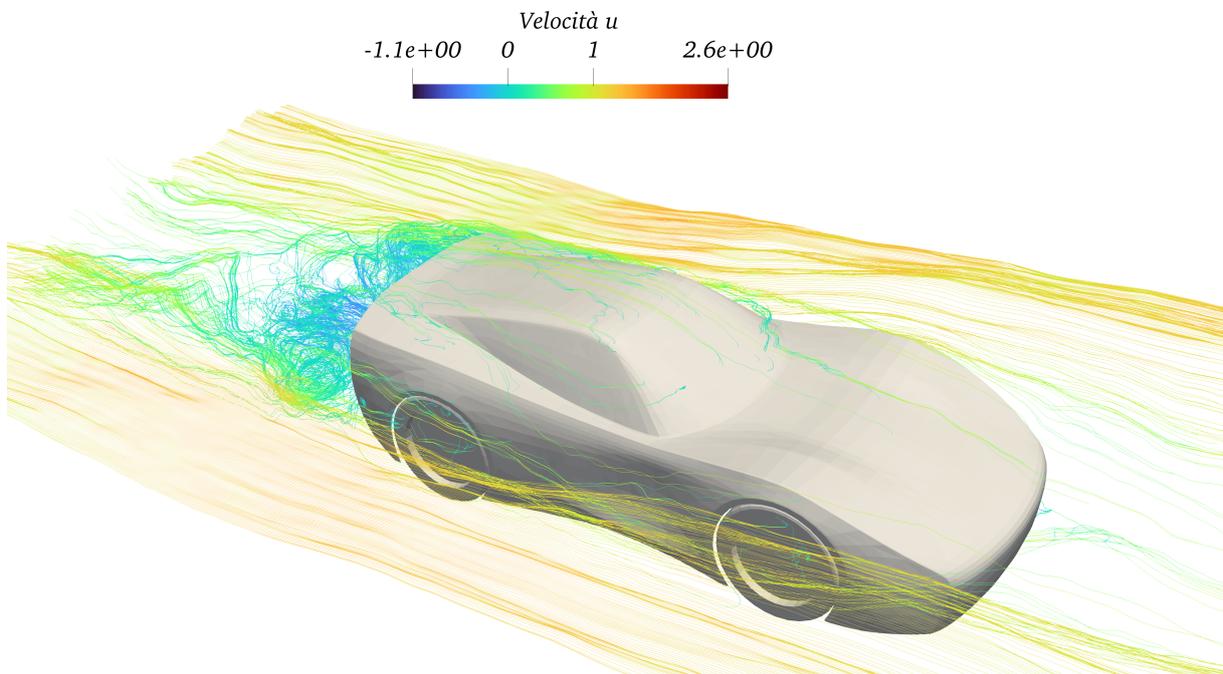
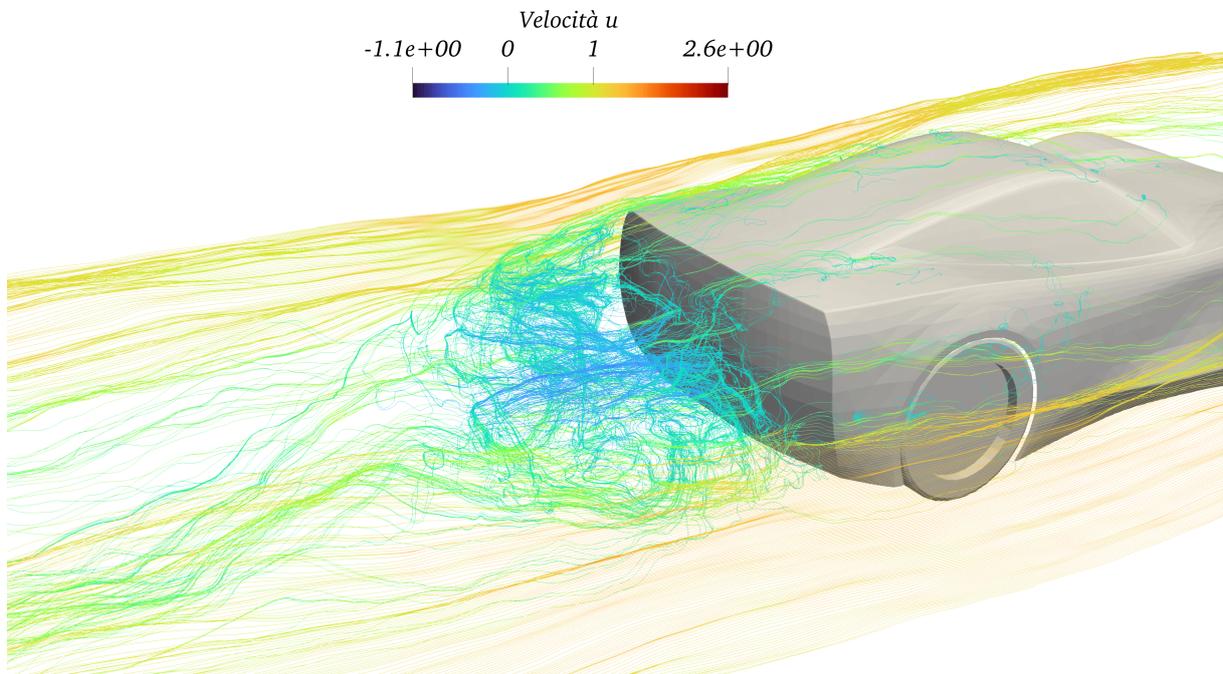


Figura 7.4: Dettaglio *streamlines* per $t = t_4$.

7.2 Flusso in Ambiente Urbano

A titolo di esempio viene simulato in un flusso in un ambiente urbano, il quale è stato sviluppato come una serie di sei parallelepipedi di diverse dimensioni all'interno del canale. Anche in questo caso è stata calcolato il SDF della geometria: In Figura 7.5 viene disegnato il *contour plot* per il valore di $SDF = 0$, congruente con la geometria STL di partenza.

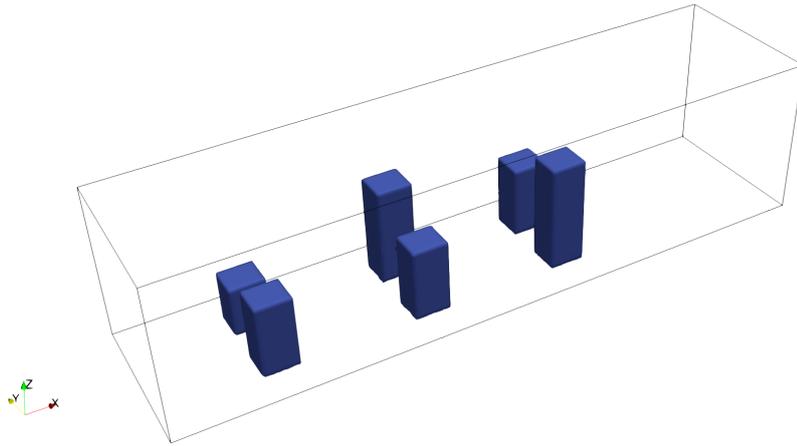
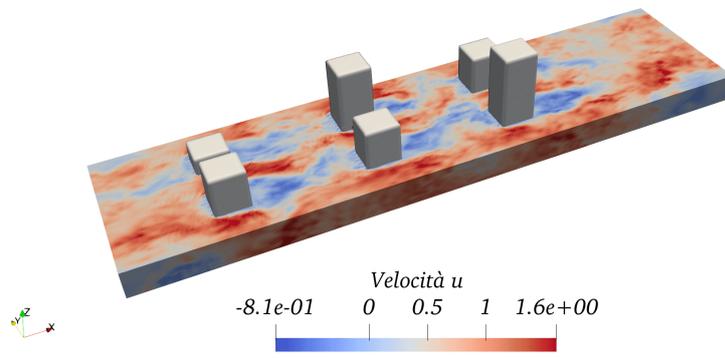
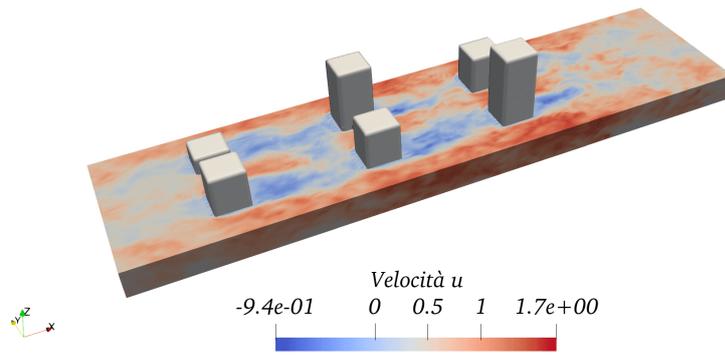


Figura 7.5: *Contout Plot* per il valore di $SDF = 0$.

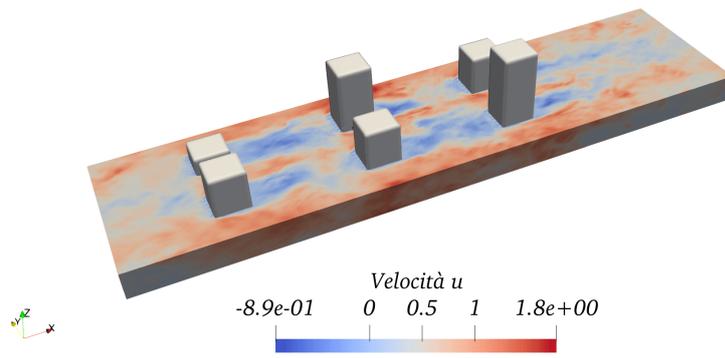
La simulazione viene condotta su un dominio di dimensioni $L_x \times L_y \times L_z = 4 \times 1 \times 1$ e con un numero di punti pari a $n_x \times n_y \times n_z = 256 \times 64 \times 64$; il numero di Reynolds viene portato a $Re = 20000$. In questo caso vengono scelte delle condizioni al contorno periodiche sulle facce di ingresso e uscita del fluido lungo l'asse x . In Figura 7.6 viene raffigurato l'andamento della velocità u , ovvero lungo la coordinata x , per quattro tempi differenti. Inoltre in Figura 7.7, per il tempo $t = 3$ viene rappresentata una sezione del campo di velocità e, al fine di illustrare la tridimensionalità del fenomeno simulato, vengono riportate le tre componenti della velocità u , v e w .



(a) $t = 1.$

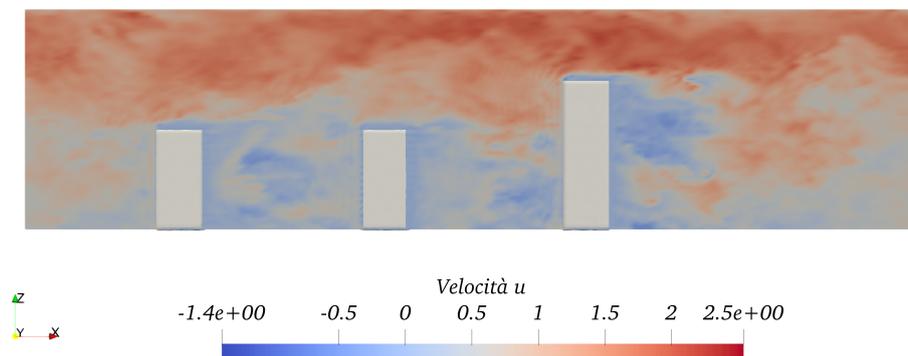


(b) $t = 2.$

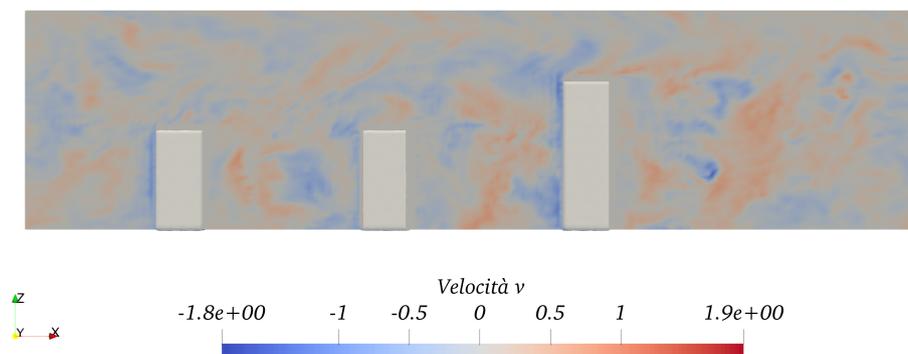


(c) $t = 3.$

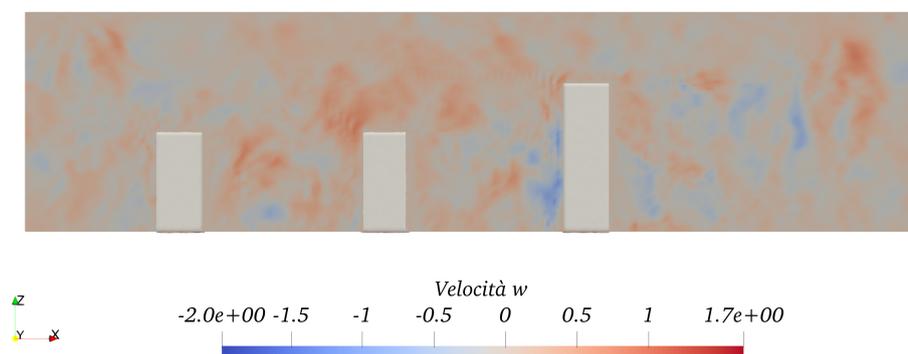
Figura 7.6: Velocità u a tre tempi differenti.



(a) Componente u .



(b) Componente v .



(c) Componente w .

Figura 7.7: Le tre componenti della velocità per $t = 3$.

Capitolo 8

Conclusioni

Il principale obiettivo di questa tesi è l'implementazione di un *Signed Distance Field* in accoppiamento con una tecnica *Immersed Boundary Method* simulando i flussi turbolenti con una *Large Eddy Simulation* modellata a parete al fine di gestire delle simulazioni di flussi ingegneristici che investono geometrie complesse. La validazione, condotta su due casi e confrontata con i dati disponibili in letteratura, ha dimostrato che il codice sviluppato offre dei risultati accurati attraverso simulazioni stabili e, grazie alla possibilità di parallelizzare i processi, con tempi di calcolo limitati.

Testare il codice su alcune geometrie complesse ha messo in luce le sue potenzialità, dimostrando come il *Signed Distance Field* risulti un ottimo metodo per la rappresentazione della geometria in un dominio computazionale e come funzioni in modo accurato in accoppiamento con una tecnica *Immersed Boundary Method*.

Sviluppi futuri di questo metodo possono riguardare l'accoppiamento del *Signed Distance Field* con tecniche *Immersed Boundary Method* differenti, al fine di individuare una combinazione che possa ridurre ulteriormente i tempi di calcolo, oppure implementare l'utilizzo del presente metodo su geometrie in movimento.

Bibliografia

- [1] Richard P Beyer e Randall J LeVeque. «Analysis of a one-dimensional model for the immersed boundary method». In: *SIAM Journal on Numerical Analysis* 29.2 (1992), pp. 332–364.
- [2] Wim-Paul Breugem. «A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows». In: *Journal of Computational Physics* 231.13 (2012), pp. 4469–4498.
- [3] Haecheon Choi e Parviz Moin. «Grid-point requirements for large eddy simulation: Chapman’s estimates revisited». In: *Physics of fluids* 24.1 (2012).
- [4] Pedro Costa. «A FFT-based finite-difference solver for massively-parallel direct numerical simulations of turbulent flows». In: *Computers & Mathematics with Applications* 76.8 (2018), pp. 1853–1862.
- [5] SCR Dennis e Gau-Zu Chang. «Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100». In: *Journal of Fluid Mechanics* 42.3 (1970), pp. 471–489.
- [6] Jacob E Fromm e Francis H Harlow. «Numerical solution of the problem of vortex street development». In: *The Physics of Fluids* 6.7 (1963), pp. 975–982.
- [7] Richard A Gentry, Robert E Martin e Bart J Daly. «An Eulerian differencing method for unsteady compressible flow problems». In: *Journal of computational Physics* 1.1 (1966), pp. 87–118.
- [8] D Goldstein, R Handler e L Sirovich. «Modeling a no-slip flow boundary with an external force field». In: *Journal of computational physics* 105.2 (1993), pp. 354–366.

- [9] TA Johnson e VC Patel. «Flow past a sphere up to a Reynolds number of 300». In: *Journal of Fluid Mechanics* 378 (1999), pp. 19–70.
- [10] Marian Kleineberg. *mesh-to-sdf*. Ver. 0.0.15. 2020. URL: https://github.com/marian42/mesh_to_sdf.
- [11] Andrey Nikolaevich Kolmogorov. «Dissipation of energy in the locally isotropic turbulence». In: *Dokl. Akad. Nauk. SSSR*. Vol. 32. 1941, pp. 19–21.
- [12] Pijush K Kundu, Ira M Cohen e David R Dowling. *Fluid mechanics*. Academic press, 2015.
- [13] Ming-Chih Lai e Charles S Peskin. «An immersed boundary method with formal second-order accuracy and reduced numerical viscosity». In: *Journal of computational Physics* 160.2 (2000), pp. 705–719.
- [14] Zhengshou Lai et al. «Signed distance field enhanced fully resolved CFD-DEM for simulation of granular flows involving multiphase fluids and irregularly shaped particles». In: *Computer Methods in Applied Mechanics and Engineering* 414 (2023), p. 116195.
- [15] Mark N Linnick e Hermann F Fasel. «A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains». In: *Journal of Computational Physics* 204.1 (2005), pp. 157–192.
- [16] Haoxiang Luo. «Immersed Boundary Method». In: *Encyclopedia of Microfluidics and Nanofluidics*. A cura di Dongqing Li. Boston, MA: Springer US, 2008, pp. 805–808.
- [17] S Marella et al. «Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations». In: *Journal of Computational Physics* 210.1 (2005), pp. 1–31.
- [18] Rajat Mittal e Gianluca Iaccarino. «Immersed boundary method». In: *Annu.Rev.Fluid Mech.* 37 (gen. 2005), pp. 239–261.
- [19] Rajat Mittal e Jung Hee Seo. «Origin and evolution of immersed boundary methods in computational fluid dynamics». In: *Physical review fluids* 8.10 (2023), p. 100501.

- [20] Franck Nicoud e Frédéric Ducros. «Subgrid-scale stress modelling based on the square of the velocity gradient tensor». In: *Flow, turbulence and Combustion* 62.3 (1999), pp. 183–200.
- [21] Jeong Joon Park et al. «DeepSDF: Learning continuous signed distance functions for shape representation». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 165–174.
- [22] Charles S Peskin. «Numerical analysis of blood flow in the heart». In: *Journal of Computational Physics* 25.3 (1977), pp. 220–252.
- [23] Stephen B Pope. «Turbulent flows». In: *Measurement Science and Technology* (2001).
- [24] Marvin Rich e Samuel S Blackman. *A method for Eulerian fluid dynamics*. Vol. 2826. Los Alamos Scientific Laboratory of the University of California, 1963.
- [25] Lewis F Richardson. *Weather prediction by numerical process*. University Press, 1922.
- [26] Márta Szilvsi-Nagy e GY Matyasi. «Analysis of STL files». In: *Mathematical and computer modelling* 38.7-9 (2003), pp. 945–960.
- [27] Kunihiko Taira e Tim Colonius. «The immersed boundary method: a projection approach». In: *Journal of Computational Physics* 225.2 (2007), pp. 2118–2137.
- [28] Markus Uhlmann. «An immersed boundary method with direct forcing for the simulation of particulate flows». In: *Journal of computational physics* 209.2 (2005), pp. 448–476.
- [29] J. A. Viegli. «A computing method for incompressible flows bounded by moving walls». In: *Journal of Computational Physics* 8.1 (1971), pp. 119–143.
- [30] J.A. Viegli. «A method for including arbitrary external boundaries in the MAC incompressible fluid computing technique». In: *Journal of Computational Physics* 4.4 (1969), pp. 543–551.
- [31] JE Welch et al. «A computing technique for solving viscous incompressible transient fluid flow problems involving free-surfaces». In: *Los Alamos Sci. Lab., LA-3425* (1966).

- [32] Tao Ye et al. «An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries». In: *Journal of computational physics* 156.2 (1999), pp. 209–240.
- [33] Chenguang Zhang. «sdfibm: a signed distance field based discrete forcing immersed boundary method in OpenFOAM». In: *Computer Physics Communications* 255 (2020), p. 107370.
- [34] Chenguang Zhang, Chunliang Wu e Krishnaswamy Nandakumar. «Effective geometric algorithms for immersed boundary method using signed distance field». In: *Journal of Fluids Engineering* 141.6 (2019), p. 061401.