



**Politecnico
di Torino**

Politecnico di Torino

Ingegneria Aerospaziale

A.A. 2021/2022

Sessione di Laurea luglio 2022

A Comparison of different Immersed Boundary Methods for Incompressible Solver

Relatori:

Francesco Larocca
Andrea Ferrero
Kostantinos Samouchos
Marco Cisternino
Andrea Iob

Candidato:

Zeffiro Soldi

Indice

Introduzione.....	9
Immersed Boundary Method.....	11
2.1 Equazioni di governo.....	11
2.2 Generazione della griglia.....	12
2.2.1 Superficie del Corpo	12
2.2.2 Rifiamento della Griglia.....	13
2.3 Condizioni al Contorno	15
2.3.1 Approccio Continuous Forcing	15
2.3.2 Approccio Discrete Forcing	16
ImmerFlow	19
3.1 Equazioni di Governo e Discretizzazione	19
3.2 Celle di Confine e Impermeabilità.....	20
3.3 Condizioni al contorno	20
3.3.1 Termine Convettivo	21
3.3.2 Termine Diffusivo.....	21
I metodi di Interpolazione.....	22
4.1 Origini.....	22
4.2 Descrizione del Solutore.....	24
4.3 Schemi di Interpolazione.....	24
4.3.1 Interpolazione Lineare o Standard Reconstruction Method.....	25
4.3.2 Interpolazione Lineare con Gradiente di Pressione	26
4.3.3 Interpolazione Quadratica.....	27
Schemi di Interpolazione.....	29
5.1 Interpolazione Lineare	29
5.2 Interpolazione Multi Lineare.....	30
5.2.1 Convezione	30
5.2.2 Diffusione	32
5.3 Interpolazione Multi Lineare con Gradiente di Pressione	35
5.4 Interpolazione Quadratica	35
5.4.1 Convezione	35
5.4.2 Diffusione	37
Risultati Numerici	39
6.1 Cilindro Circolare Infinito Laminare	39
6.1.1 Descrizione del caso	39

6.1.2 Alcuni risultati.....	41
6.2 Lamina Piana Laminare.....	49
6.2.1 Lamina Orizzontale.....	50
6.2.2 Lamina Inclinata.....	58
6.3 Condotto Tridimensionale Laminare.....	65
6.3.1 Descrizione del caso	65
6.3.2 Alcuni Risultati	67
Conclusioni.....	71
Appendice A: Staggered Grid	74

Sommario

L'elaborato qui presentato si pone l'obiettivo di implementare una serie di nuove metodologie per risolvere il campo di moto all'interfaccia di separazione corpo fluido in un solutore fluidodinamico ai volumi finiti basato sul metodo *Immersed Boundary*.

Il lavoro, svolto presso l'azienda Optimad s.r.l, si è suddiviso in due parti. La prima parte ha riguardato la ricerca bibliografica di materiale: innanzitutto, per comprendere al meglio la filosofia *Immersed Boundary*, la generazione di griglie cartesiane, tecniche di affinamento in stile *octree* e, per ultimo, nuove strategie di risoluzioni delle equazioni di *Navier-Stokes* nei pressi di superfici di separazione.

Nella seconda parte, ha visto come protagonista l'implementazione, in linguaggio C++, di quanto visto durante la fase di ricerca bibliografica.

Partendo dall'idea di Fadlun (Fadlun, Verzicco, Orlandi, Mohd-Yusof, “*Combined Immersed Boundary Finite Difference Methods for Three-Dimensional Complex Flow Simulation*”, Journal of Computational Physics 161, 35–60, 2000), rivista successivamente da Iaccarino (Kang, Iaccarino, Moin, “*Accurate Immersed Boundary Reconstruction for Viscous Flow Simulations*”, AIAA Journal vol. 47 n.7, 1750-1760, 2009), i quali presentarono risultati molto buoni in campo laminare con anche discreti risultati in campo turbolento, sono state implementate quattro tipologie di interpolazione attraverso cui si cerca di migliorare l'imposizione della condizione al contorno alla parete solida per il campo di velocità.

Nel testo vengono riportati lo studio alla base dei metodi di interpolazione per fluidi incomprimibili. Inoltre, i risultati di questo lavoro attraverso un confronto delle soluzioni ottenute tramite le modifiche apportate ad *ImmerFlow*, il precedente solutore inalterato e, infine, i risultati numerici o sperimentali, cercando di determinare in modo critico quale versione sia la migliore. Lo studio dei risultati è svolto su alcuni casi comuni come lamina piana, cilindro circolare e un condotto tridimensionale.

Abstract

The paper presented here aims to implement a set of new methodologies for solving the momentum field at the fluid-body separation interface in a finite-volume fluid dynamic solver based on the Immersed Boundary method.

The work, carried out at the company Optimad s.r.l, was divided into two parts. The first part involved a literature search for material: first, to better understand the Immersed Boundary philosophy, the generation of Cartesian grids, octree-style refinement techniques, and, lastly, new strategies for solving the Navier-Stokes equations near separation surfaces.

The second part featured the implementation, in C++ language, of what was seen during the literature search phase.

Building on the idea of Fadlun (Fadlun, Verzicco, Orlandi, Mohd-Yusof, “*Combined Immersed Boundary Finite Difference Methods for Three-Dimensional Complex Flow Simulation*”, Journal of Computational Physics 161, 35–60, 2000), later revised by Iaccarino (Kang, Iaccarino, Moin, “*Accurate Immersed Boundary Reconstruction for Viscous Flow Simulations*”, AIAA Journal vol. 47 n.7, 1750-1760, 2009), who presented very good results in the laminar field with also discrete results in the turbulent field, four types of interpolation were implemented through which an attempt is made to improve the imposition of the boundary condition on the solid wall for the velocity field.

The study behind the interpolation methods for incompressible fluids are reported in the text. Also, the results of this work through a comparison of the solutions obtained through the modifications made to ImmerFlow, the previous unaltered solver, and finally the numerical or experimental results, trying to critically determine which version is the best. The study of the results is carried out on some common cases such as flat foil, circular cylinder and a three-dimensional duct.

Indice delle Figure

Figura 1: confronto tra griglia BF e IB [2].	9
Figura 2: un esempio di rappresentazione staircase della geometria [17].	13
Figura 3: esempio di AMR con griglia ad octree [19].	14
Figura 4: esempio di rappresentazione ad albero della gerarchia [19].	14
Figura 5: nodi Lagrangiani (a sinistra). Smoother distribution function (a destra) [2].	16
Figura 6: ghost cell approach. Le lettere indicano i centri cella. 'G' ghost cell [2].	17
Figura 7: interpolazione a grado zero [1].	23
Figura 8: interpolazione pesata [1].	23
Figura 9: interpolazione lineare [1].	24
Figura 10: stencil lineare [3].	25
Figura 11: volume di controllo [3].	26
Figura 12: quadratic stencil [3].	28
Figura 13: stencil di interpolazione lineare.	30
Figura 14: stencil di interpolazione multi lineare.	31
Figura 15: rappresentazione delle aree interne allo stencil.	32
Figura 16: un triangolo generico.	33
Figura 17: piramide generica.	34
Figura 18: stencil di interpolazione parabolica.	36
Figura 19: stencil in un caso particolare.	37
Figura 20: stencil modificato per interpolazione parabolica.	37
Figura 21: dominio computazionale e griglia cilindro bidimensionale.	40
Figura 23: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, ad un diametro di distanza dal cilindro, al variare della coordinata adimensionale y/d.	41
Figura 24: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, a due diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d.	41
Figura 25: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, a quattro diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d.	42
Figura 26: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, a sette diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d.	42
Figura 27: confronto del campo di pressione attorno al cilindro. Sopra: schema multi lineare, sotto: schema multi lineare con correzione del gradiente di pressione.	43
Figura 28: andamento dei residui dell'equazione di quantità di moto lungo l'asse x.	44
Figura 29: andamento dei residui dell'equazione di quantità di moto lungo l'asse y e andamento dei residui dell'equazione di quantità di moto lungo l'asse x.	44
Figura 30: andamento dei residui dell'equazione di continuità.	45
Figura 31: schema di interpolazione lineare.	45
Figura 32: schema di interpolazione multi lineare.	46
Figura 33: schema di interpolazione quadratica.	46
Figura 34: schema di interpolazione multi lineare con correzione tramite gradiente di pressione.	47
Figura 35: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, ad un diametro di distanza dal cilindro, al variare della coordinata adimensionale y/d.	48
Figura 36: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, a due diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d.	48
Figura 37: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, a quattro diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d.	49
Figura 38: distribuzione della velocità lungo l'asse x, adimensionalizzato con la velocità all'infinito, a sette diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d.	49
Figura 39: dominio computazionale e griglia lamina piana orizzontale.	50
Figura 40: celle rimosse.	51

Figura 41: profilo a $x = 0.1$ m dal punto di attacco.	51
Figura 42: profilo a $x = 0.2$ m dal punto di attacco.	52
Figura 43: profilo a $x = 0.4$ m dal punto di attacco.	52
Figura 44: profilo a $x = 0.7$ m dal punto di attacco.	53
Figura 45: andamento dei residui dell'equazione di quantità di moto lungo l'asse x.	54
Figura 46: andamento dei residui dell'equazione di quantità di moto lungo l'asse y.	54
Figura 47: andamento dei residui dell'equazione di continuità.	55
Figura 48: schema di interpolazione lineare.	56
Figura 49: schema di interpolazione multi lineare.	56
Figura 50: schema di interpolazione quadratico.	57
Figura 51: schema di interpolazione multi lineare con correzione tramite gradiente di pressione.	57
Figura 52: griglia computazionale.	58
Figura 53: celle rimosse.	59
Figura 54: profilo a $x = 0.1$ m dal punto di attacco.	60
Figura 55: profilo a $x = 0.2$ m dal punto di attacco.	60
Figura 56: profilo a $x = 0.4$ m dal punto di attacco.	61
Figura 57: profilo a $x = 0.7$ m dal punto di attacco.	61
Figura 58: andamento dei residui dell'equazione di quantità di moto lungo l'asse x.	62
Figura 59: andamento dei residui dell'equazione di quantità di moto lungo l'asse y.	62
Figura 60: andamento dei residui dell'equazione di continuità.	63
Figura 61: schema di interpolazione lineare.	63
Figura 62: schema di interpolazione multi lineare.	64
Figura 63: schema di interpolazione quadratico.	64
Figura 64: schema di interpolazione multi lineare con correzione tramite gradiente di pressione.	65
Figura 65: dettaglio della griglia totale in un piano normale all'asse del condotto.	66
Figura 66: dettaglio della griglia nella regione fluida del condotto. La sezione è presa normale all'asse del canale.	66
Figura 67: geometria del tubo analizzato.	67
Figura 68: profilo in $x = 0.2$ m.	68
Figura 69: profilo in $x = 0.25$ m.	68
Figura 70: residui dell'equazione di quantità di moto lungo l'asse x.	69
Figura 71: residui dell'equazione di quantità di moto lungo l'asse y.	69
Figura 72: residui dell'equazione di quantità di moto lungo l'asse z.	70
Figura 73: residui dell'equazione di continuità.	70
Figura A.74: griglia staggered.	74

Indice delle Tabelle

Tabella 1: grandezze passive aria.	39
Tabella 2: posizione dei profili nel caso di lamina orizzontale.	51
Tabella 3: posizione dei profili nel caso di lamina piana inclinata di 15deg.	59
Tabella 4: posizione dei profili campionati nel condotto tridimensionale.	67

Capitolo 1

Introduzione

Una classificazione dei metodi di risoluzioni adottati dalla fluidodinamica computazionale riguarda il tipo di *griglia* adottata per svolgere le analisi numeriche:

1. Metodo *Body Fitted (BF)*.
2. Metodo *Immersed Boundary (IMB)*.

La prima, e sostanziale, differenza che distingue i due metodi è la tipologia di griglia adottata per la discretizzazione delle equazioni del moto: il primo metodo, come suggerito dal nome, impiega una griglia, strutturata o non strutturata, che si adatta al corpo, seguendone l'andamento della superficie esposta al flusso, interno o esterno che esso sia, suddividendo in volumi finiti solo la regione fluida. Il secondo metodo utilizza una griglia cartesiana che, tranne in casi particolari, non è conforme alla geometria analizzata, ma discretizza tutto il dominio computazionale indistintamente che sia interno o esterno al fluido. Questa breve descrizione evidenzia intrinsecamente un primo vantaggio che il metodo *Immersed Boundary* ha sul metodo *Body Fitted*: una più semplice e rapida generazione della griglia che in alcun modo deve conoscere particolari geometrici del corpo; ma anche uno svantaggio: la griglia raramente si adatterà al corpo, il contorno del corpo è tagliato dai volumi che compongono la griglia, questo genererà alcune problematiche, tra le quali una più difficile imposizione delle condizioni al contorno. Il presente elaborato è focalizzato su questa problematica.

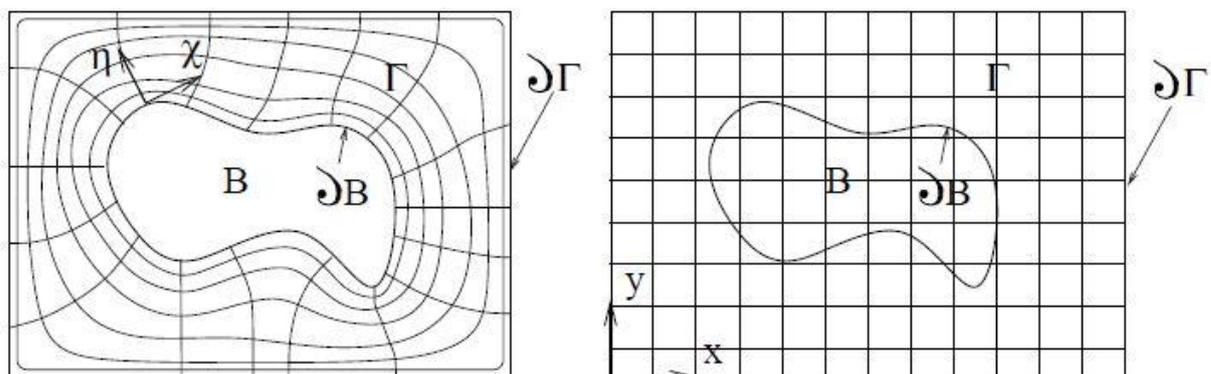


Figura 1: confronto tra griglia BF e IB [2].

Diversi sono i vantaggi e gli svantaggi che entrambi i metodi presentano, nel seguito di questa breve introduzione se ne analizzeranno alcuni dei principali. In seguito, si introdurrà il metodo di ricostruzione della soluzione implementato durante l'attività di tesi.

Come anticipato poco sopra, il primo svantaggio che la metodologia *Immersed Boundary* presenta riguarda l'imposizione delle condizioni al contorno. A causa della presenza di una griglia totalmente scollegata alla geometria, l'implementazione delle condizioni al contorno risulta difficile e poco chiara, e purtroppo non sempre verificata. Di conseguenza, vengono anche complicate e

infiata l'accuratezza della soluzione e il rispetto delle leggi di conservazione nei pressi del contorno.

D'altro canto, nel caso di risoluzioni con metodo *Body Fitted*, le sopra elencate tre problematiche sono meno marcate, presentando maggior accuratezza e dettaglio nella descrizione della geometria grazie appunto alla maggior flessibilità della griglia nell'adattarsi al contorno.

“Si consideri un problema bidimensionale con corpo caratteristico di dimensione L e strato limite di dimensione δ . Immaginando di discretizzare il dominio con una griglia di dimensione Δ_n e Δ_t nelle direzioni normali e tangenziali alla parete del corpo. Per moderati numeri di Reynolds $\delta \ll L$, il rapporto tra dimensioni geometriche e la dimensione di griglia per griglie conformi tradizionalmente è $(L/\Delta_t)(\delta/\Delta_t)$, mentre per una griglia cartesiana è L^2/Δ_n^2 . Assumendo $\Delta_t \propto L$ e $\Delta_n \propto \delta$, si ottiene un rapporto di griglia di tra i due metodi di $(L/\delta)^2$. Assumendo, inoltre, un regime laminare per cui $L/\delta \propto Re^{0.5}$, il rapporto delle griglie cresce proporzionale al numero di Reynolds, per casi 2D, e $Re^{1.5}$, per casi 3D. Perciò, il numero di celle cresce più rapidamente per i metodi *immersed-boundary*, ma questo non implica per forza un aumento del costo computazionale perché una alta frazione dei nodi potrebbe risultare interna al corpo e quindi non risolta.” (Mittal, Iaccarino, “*Immersed Boundary Methods*”, Annual Review of Fluid Mechanics 37, 239-261, 2005, et al [2])

Il vantaggio sostanziale che il metodo *Immersed Boundary* presenta su di un metodo *Body Fitted* è l'estrema semplicità di generazione della griglia. La generazione della griglia è perciò decisamente semplificata e rapida. Infatti, generare una griglia conforme anche per geometrie semplici, mantenendo alta la risoluzione a parete con il minor numero di elementi possibile, risulta essere spesso un procedimento iterativo e richiedente diversi input da parte dell'utente nel caso di griglia *Body Fitted*. Al contrario per un metodo *Immersed Boundary*, la complessità e qualità della griglia non risentono particolarmente della complessità della geometria.

Particolare attenzione è da porre in casi in cui si vuole risolvere un campo di moto presentante corpi in movimento: se si utilizza una griglia conforme bisognerà ad ogni istante generare una nuova griglia e successivamente proiettare la soluzione dalla vecchia griglia, compromettendo la semplicità, accuratezza, robustezza e il costo computazionale. Al contrario, quando si utilizza una discretizzazione con metodo *Immersed Boundary*, la griglia può rimanere fissa, permettendo di evitare le problematiche esposte.

Capitolo 2

Immersed Boundary Method

In questo capitolo, viene presentata una rapida panoramica sui metodi *Immersed Boundary*. In particolare, si discuterà a riguardo delle equazioni di governo, della creazione della griglia per la discretizzazione del dominio computazionale e, infine, vengono presentati alcuni metodi di forcing per la soluzione delle equazioni discrete in prossimità dei contorni.

2.1 Equazioni di governo

Come per tutte le tecniche di risoluzione per la fluidodinamica, l'obiettivo dei metodi *Immersed Boundary* è quello di risolvere le equazioni di *Navier-Stokes*, le quali per un fluido newtoniano sono:

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \mathbf{V} \cdot \nabla(\rho \mathbf{V}) = 0 \\ \rho \frac{\partial \mathbf{V}}{\partial t} + \rho \mathbf{V} \cdot \nabla \mathbf{V} = -\nabla p + \mu \nabla^2 \mathbf{V} + \frac{1}{3} \nabla(\nabla \cdot \mathbf{V}) + \mathbf{f} \\ \rho \frac{\partial E}{\partial t} + \rho \mathbf{V} \cdot \nabla E = -\nabla \cdot (p \mathbf{V}) + \nabla \cdot (\bar{\mathbf{t}} \cdot \mathbf{V}) + \rho \mathbf{f} \cdot \mathbf{V} - \nabla \cdot \mathbf{q} + q_v \end{array} \right. \quad 2.1$$

Queste equazioni, nel caso più generale, forniscono la soluzione a qualsiasi campo di moto di fluidi sull'intero dominio fluido Ω_f .

Per questo elaborato si considereranno solamente i fluidi incomprimibili, caratterizzati da avere la densità costante in tutto il dominio fluido, per i quali l'equazione di compatibilità può essere riscritta attraverso la condizione di solenoidalità del campo di velocità:

$$\nabla \cdot \mathbf{V} = 0 \quad \text{in } \Omega_f \quad 2.2$$

Inoltre, per i fluidi incomprimibili il campo di moto della velocità e il campo di energia non sono legati, in quanto la viscosità non è funzione della temperatura del fluido, portando ad una semplificazione del sistema matematico originale perché l'equazione della temperatura può essere risolta a seguito dell'ottenimento della soluzione per il campo di velocità. Si considererà perciò nel seguito solo l'equazione della quantità di moto, riscritta per i flussi incomprimibili:

$$\rho \frac{\partial \mathbf{V}}{\partial t} + \rho \mathbf{V} \cdot \nabla \mathbf{V} = -\nabla p + \mu \nabla^2 \mathbf{V} \quad \text{in } \Omega_f \quad 2.3$$

Il primo metodo *Immersed Boundary*, metodo dei confini immersi, è nato dall'idea di Perkins (1972) per lo studio e simulazione del flusso di sangue in cuore pulsante, con geometria elastica e mobile. Lo studio utilizzava le equazioni di *Navier-Stokes* incomprimibile ed una griglia cartesiana.

Il nome del metodo trae spunto dalla visualizzazione del dominio discretizzato per la risoluzione numerica delle equazioni, in quanto il corpo risulta essere immerso nella griglia.

In generale, perciò si chiamano *Immersed Boundary* tutti quei metodi che utilizzano una griglia che non si adatta al corpo. Viene introdotta una condizione che forza determinate condizioni sulle superfici che corrispondono alla posizione fisica di geometrie complesse.

2.2 Generazione della griglia

Nota la geometria del problema, il primo passaggio per la risoluzione delle equazioni in forma numerica è la generazione della griglia computazionale, la griglia.

Per tutte le analisi numeriche la griglia non è altro che un reticolo, più o meno complesso, generato dall'intersezione di curve sul quale vengono risolte le equazioni differenziali alle derivate parziali descriventi il fenomeno fisico studiato.

Talvolta nelle analisi fluidodinamiche, in special modo per quelle svolte con tecnica *Immersed Boundary*, si ricorre a griglie di tipo *staggered*, si veda l'[appendice A](#) per maggiori dettagli.

2.2.1 Superficie del Corpo

Per quanto riguarda i metodi *Immersed Boundary*, la griglia non rappresenta un grande problema, come precedentemente introdotto il reticolo è costruito con rette nello spazio cartesiano. Le rette suddividono il dominio computazionale in cubi, che costituiscono i volumi in cui ottenere la soluzione.

Assumendo di essere in possesso della griglia finale, il prossimo passaggio è ottenere una rappresentazione approssimata del contorno della geometria. Il risultato finale è una rappresentazione a gradini detto *staircase boundary*, letteralmente confine a gradini, o *contorno virtuale*. Questo processo è svolto attraverso l'etichettatura delle celle. Il risultato di queste è la suddivisione dei volumi nei seguenti gruppi: *interni* al dominio fluido, *esterni* al dominio fluidi o di *interfaccia* parzialmente interni e parzialmente esterni. A seconda dell'autore la nomenclatura potrebbe cambiare, ma il risultato manterrebbe lo stesso valore. In altri casi, la classificazione risulta semplificata, casistiche in cui le celle di *interfaccia* sono assenti.

Diverse sono le tecniche utili a questo scopo, tra tutte la più utilizzata è conosciuta come *ray tracing*, tipicamente utilizzata in programmi di grafica. Viene individuato un punto, che deve essere assolutamente esterno alla geometria, dal quale per ogni punto della griglia si genera un raggio e, semplicemente, calcolando il numero di intersezioni tra la geometria e il raggio stesso si individua a quale categoria un nodo appartiene. Se le intersezioni sono in numero *pari* l'elemento è esterno, se sono *dispari* allora è interno. Grazie alla rappresentazione CAD e al formato STL (Stereo-Lithography), tramite il quale una qualsiasi geometria è rappresentata come semplici triangoli inversamente proporzionali al raggio di curvatura locale, l'intersezione è calcolata tra raggio e triangolo.

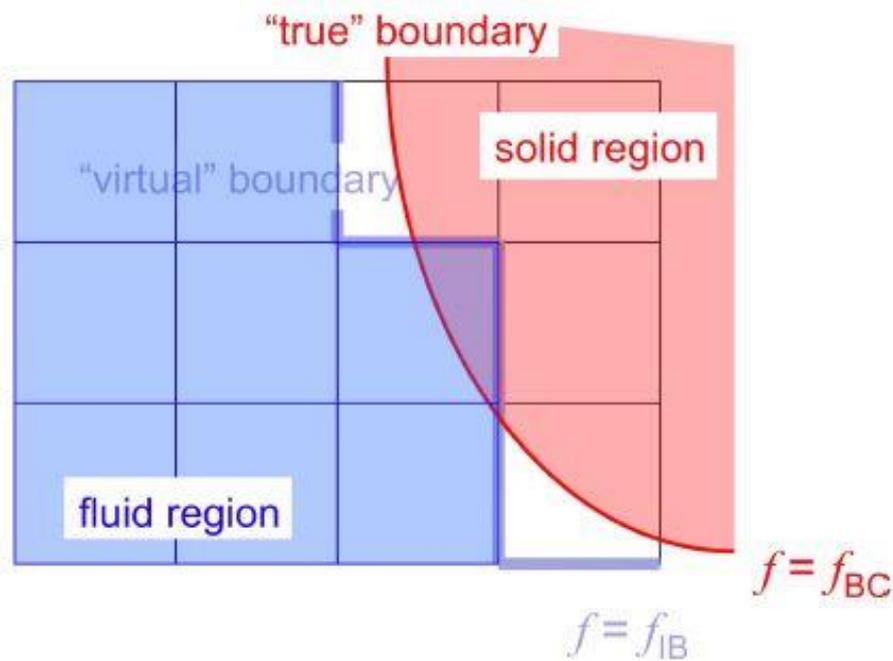


Figura 2: un esempio di rappresentazione staircase della geometria [17].

2.2.2 Rifinimento della Griglia

L'ostacolo maggiore risulta essere quello di ottenere una sufficiente risoluzione nei pressi della geometria. In breve, la griglia deve essere abbastanza fine da garantire una buona rappresentazione della variazione delle grandezze nel dominio, ma anche abbastanza rada da non causare un eccessivo aumento del costo computazionale.

Talvolta, per raggiungere gli standard di accuratezza richiesti, nei pressi della geometria, serve avere elementi estremamente piccoli e se la griglia dovesse essere uniforme su tutto il dominio potrebbe raggiungere velocemente un alto numero di elementi di griglia. Sono nati allora diversi metodi che permettano di ridurre il costo computazionale riducendo gli elementi. Uno tra i tanti, è noto come *adaptive mesh refinement*, secondo il quale a partire da una griglia di livello 0 di base, si genera una gerarchia di griglie sempre maggiormente raffinate solamente nelle zone in cui si trovano le strutture di maggior interesse. Ad esempio, si potrebbe arrivare ad una situazione che una griglia al livello n-esimo, figlia di una griglia più rada al livello (n-1)-esimo, sia anche padre di una terza più fine al livello (n+1)-esimo.

Da questa idea, è stata sviluppato un raffinamento definito *octree*. Considerando un caso tridimensionale in un sistema di riferimento cartesiano, il raffinamento può avvenire solo per bisezione dei lati di un cubo, generandone al massimo altri otto di lato la metà dell'originario. Solitamente, la gerarchia è rappresentabile attraverso una rappresentazione ad albero.

Per maggiore chiarezza, si consideri l'esempio bidimensionale seguente: partendo dalla griglia base al livello 0, costituita da 4x4 volumi, e ipotizzando che questo grado di definizione non sia sufficiente, per bisezione, con il metodo *octree*, si otterranno altre quattro griglie al livello 1 tutte composte da 4x4 celle, con lato la metà dei quadrati del livello precedente. A questo punto il procedimento è iterativo: ogni qual volta una zona debba essere rifinita verranno create quattro nuovi settori con 4x4 volumi che si sovrapporranno ai livelli precedentemente creati. Nella figura 3 e nella figura 4 si può visualizzare l'esempio appena riportato e la correlata rappresentazione ad albero.

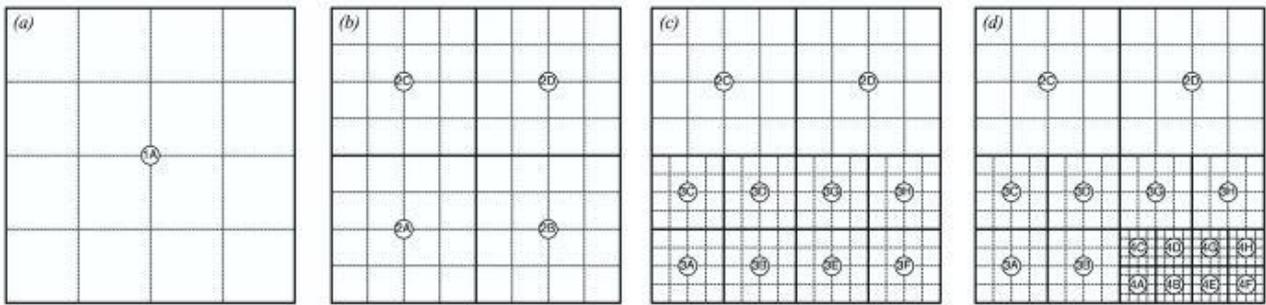


Figura 3: esempio di AMR con griglia ad octree [19].

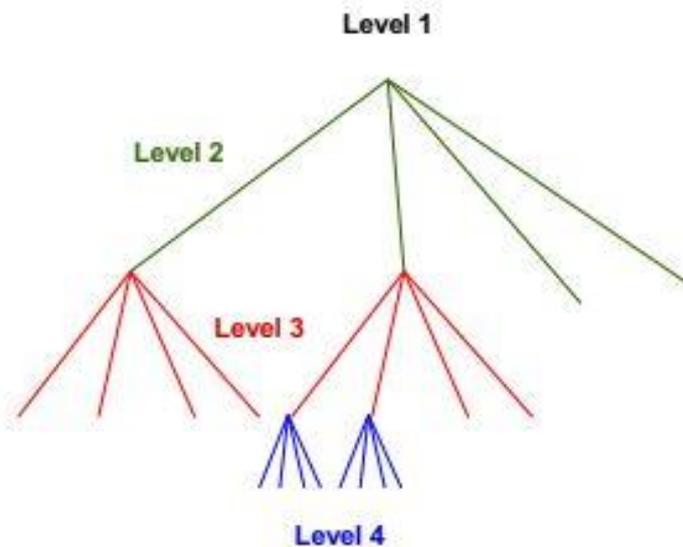


Figura 4: esempio di rappresentazione ad albero della gerarchia [19].

Diversi possono essere i criteri per cui può avvenire il affinamento, alcuni sono qui sottoelencati:

- per considerazioni geometriche: si mantengono celle più fini nei pressi della geometria o altre zone di interesse. Oppure, si sceglie di mantenere la griglia quanto più rada possibile all'esterno del dominio fluido, riducendo le informazioni da memorizzare al computer.
- per volontà dell'utente: il quale decide che nell'immediata vicinanza alla geometria ci debba essere un determinato numero di celle della stessa dimensione.
- a seguito di un'analisi a posteriori della soluzione: tramite la definizione di alcuni "sensori" che, calcolando alcuni parametri del flusso, sono in grado di capire se una cella abbia bisogno di affinamento. Alcuni di questi possono essere, ad esempio, la divergenza della velocità, il gradiente di pressione, la vorticità e il gradiente dell'energia turbolenta.

Un ulteriore vantaggio dell'*adaptive mesh refinement* è la bi-direzionalità del processo: ogni qual volta una zona non risulta più di interesse può essere de-rifinita ingrandendo la dimensione della cella. Qui tornano utili i "sensori" appena definiti, infatti, questi possono anche informare il solutore che è possibile unire celle per formare una di dimensione maggiore.

A questo punto, una volta modificato il reticolo devono essere trasportate le variabili dalla cella padre alla cella figlia (dalla cella figlia alla cella padre). Queste operazioni prendono il nome di *prolongation e restriction* [19], solitamente sono operazioni di interpolazione delle variabili da celle limitrofe, della forma:

$$I(\varphi)_{i',j',k'}^{l+1} = \sum_{p,q,r=il}^{el} \alpha_{p,q,r}(\varphi)_{i+p,j+q,k+r}^l \quad 2.4$$

per l'operatore di *prolongation*, e

$$R(\varphi)_{i,j,k}^l = \sum_{p,q,r=il}^{el} \beta_{p,q,r}(\varphi)_{i+p,j+q,k+r}^{l+1} \quad 2.5$$

per l'operatore di *restriction*.

Questi operatori non garantiscono sempre la solenoidalità del campo di velocità quando avviene il passaggio delle informazioni tra due zone del dominio a diversi livelli di griglia.

Per maggiori informazioni a riguardo la generazione di griglie e raffinamento fare riferimento a [5, 7, 12, 13, 19].

2.3 Condizioni al Contorno

Il problema principale riguarda l'imposizione delle condizioni al contorno. Diversi sono stati gli studi per cercare di risolvere il problema. Solitamente, le condizioni al contorno sono imposte attraverso la modifica delle equazioni, imponendo indirettamente in precisi punti sulla griglia. Per flussi incomprimibili importanti sono gli approcci noti come *continuos forcing approach* e *direct forcing approach* [2]. Entrambi vedono l'introduzione di un termine forzante di varia natura.

2.3.1 Approccio Continuos Forcing

Semplificando il sistema di equazioni sopra riportato:

$$[L]\{\mathbf{U}\} = \{0\} \text{ in } \Omega_f, \quad \text{con } \{\mathbf{U}\} = \{\mathbf{U}_\Gamma\} \text{ su } \Gamma_b \quad 2.6$$

L'operatore $[L]$ rappresenta le equazioni di *Navier-Stokes* e $\{\mathbf{U}\}$ il vettore soluzione composto dalle componenti di velocità e la pressione.

Si può dire che il primo approccio punta ad inserire nelle equazioni un termine forzante sull'intero dominio, ottenendo:

$$[L]\{\mathbf{U}\} = \{\mathbf{f}_b\} \text{ in } \Omega_f, \quad \text{con } \{\mathbf{U}\} = \{\mathbf{U}_\Gamma\} \text{ su } \Gamma_b \quad 2.7$$

La scelta del termine forzante è dettata dal problema, in particolare dal tipo contorno. Infatti, questo approccio è nato per risolvere problemi che presentano confini del dominio elastici, come possono essere flussi biologici e multifase, e poi successivamente è stato anche esteso a domini rigidi.

L'idea iniziale era di trattare il dominio immerso come delle fibre elastiche, la cui posizione era tracciata attraverso una collezione di punti, *nodi Lagrangiani*, in moto alla velocità del flusso. Vengono calcolate deformazioni e stress delle fibre attraverso la legge di Hooke. L'effetto delle fibre è trasmesso al fluido tramite termini sorgente localizzati nel flusso con equazioni della forma:

$$\mathbf{f}_m(\mathbf{x}_{i,j}, t) = \sum_k \mathbf{F}_k(t) d(|\mathbf{x}_{i,j} - \mathbf{x}_k|) \quad 2.8$$

Dove \mathbf{x}_k rappresenta la posizione originaria della k -esima fibra, $\mathbf{F}_k(t)$ lo stress calcolato dalla legge di Hooke. Solitamente i nodi Lagrangiani non coincidono con i nodi cartesiani, allora attraverso la *smoother distribution function*, d , la forzante viene ridistribuita alle celle limitrofe al corpo.

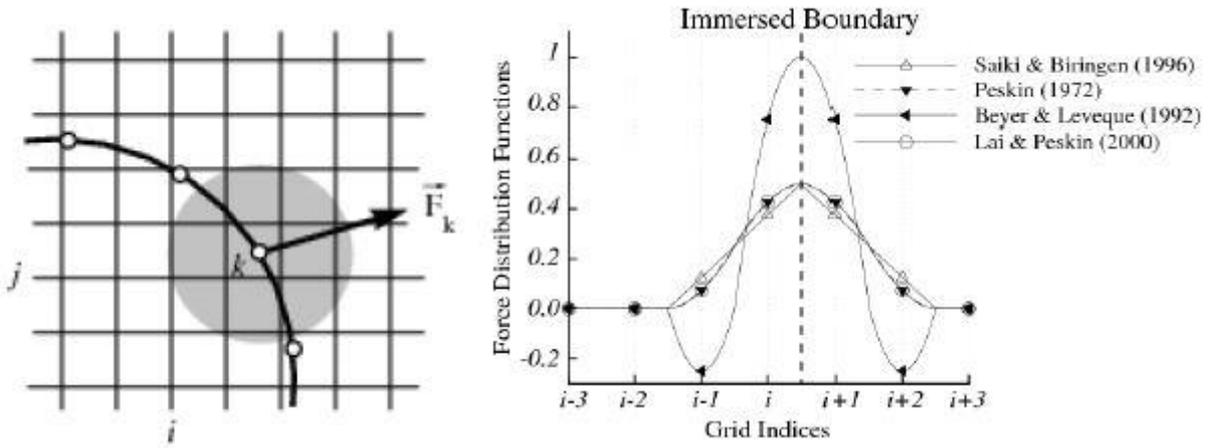


Figura 5: nodi Lagrangiani (a sinistra). Smoother distribution function (a destra) [2].

Nel caso di corpi rigidi, è stata introdotta la struttura come in equilibrio sotto l'effetto di una molla centrata nel nodo Lagrangiano. Data la rigidità del corpo la costante elastica k dovrebbe assumere valori molto elevati causando un irrigidimento del sistema di equazioni, il quale sarebbe soggetto a stringenti vincoli di stabilità.

Altre tipologie di forzanti per i corpi rigidi sono state introdotte, come per esempio:

$$\mathbf{F}(t) = \alpha \int_0^t \mathbf{u}(\tau) d\tau + \beta \mathbf{u}(t) \quad 2.9$$

Le costanti α e β sono arbitrarie e possono essere scelte in modo da rispettare al meglio le condizioni al contorno.

2.3.2 Approccio Discrete Forcing

Il secondo approccio vede in un primo momento discretizzate le equazioni del moto su tutto il dominio senza rispetto del contorno immerso, in seguito, la discretizzazione delle celle in prossimità al corpo è aggiustata per tenere in conto delle condizioni al contorno, risultante in un sistema di equazioni:

$$[L']\{\mathbf{U}\} = \{\mathbf{r}\} \quad 2.10$$

Dove \mathbf{r} risulta essere un termine noto associato alle condizioni al contorno. Quest'ultimo sistema può essere riscritto nella forma:

$$[L]\{\mathbf{U}\} = \{\mathbf{f}'_b\} \text{ dove } \{\mathbf{f}'_b\} = \{\mathbf{r}\} + [L]\{\mathbf{U}\} - [L']\{\mathbf{U}\} \quad 2.11$$

Anche questo 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 la l'eliminazione dei vincoli di stabilità.

2.3.2.1 Metodi Indiretti

Tranne che in 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. Sono stati sviluppati alcuni metodi che riescono a ricostruire la forzante tramite stime a priori.

Il primo passo è la risoluzione delle equazioni senza il rispetto delle condizioni al contorno:

$$[L]\{\mathbf{U}'\} = 0 \quad 2.12$$

ottenendo la soluzione predittiva $\{\mathbf{U}'\}$. A questo punto la forzante è ricavabile da:

$$\{\mathbf{f}'_b\} \approx \{\mathbf{r}\} + [L]\{\mathbf{U}'\} - [L']\{\mathbf{U}'\} = \{\mathbf{r}\} - [L']\{\mathbf{U}'\} \quad 2.13$$

Dove:

$$\{\mathbf{r}\} = \{\mathbf{U}_\Gamma\} d(|\mathbf{X}_k - \mathbf{x}_{i,j}|) \quad 2.14$$

Un esempio è riportato è riportato in [35].

2.3.2.2 Metodi Diretti

Questi metodi permettono di risolvere campi di moto a numeri di Reynolds maggiori senza introdurre effetti dovuti alla distribuzione della forzante.

Uno di questi è l'approccio *ghost cell*. Vengono definite celle fantasma quelle celle solide che presentano almeno un'interfaccia comune con una cella fluida, qui la soluzione viene implicitamente ricostruita tramite uno schema di interpolazione in modo tale da rispettare la soluzione sulla superficie immersa. Diverse sono le possibilità che sono state adoperate, come diverso è l'ordine di interpolazione. Alcuni esempi per un caso bidimensionale possono essere un'interpolazione bilineare nelle direzioni cartesiane:

$$\Phi = C_1 x_1 x_2 + C_2 x_1 + C_3 x_2 + C_4 \quad 2.15$$

Oppure ancora, uno schema quadratico nella direzione della normale a parete e lineare nella direzione tangenziale:

$$\Phi = C_1 n^2 + C_2 n t + C_3 n + C_4 t + C_5 \quad 2.16$$

I coefficienti saranno calcolati all'occorrenza partendo dalla soluzione nelle celle fluide limitrofe e il punto più vicino sulla parete immersa.

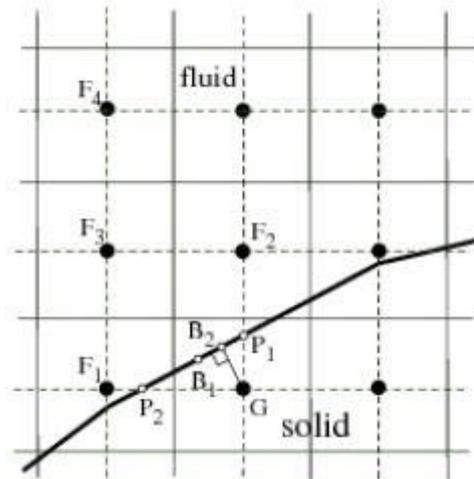


Figura 6: *ghost cell approach*. Le lettere indicano i centri cella. 'G' ghost cell [2].

Ora la soluzione nella ghost cell è esprimibile come

$$\Phi_G = \sum_{stencil} \omega_i \Phi_i \quad 2.11$$

I parametri sono funzione della geometria e ora la soluzione nella ghost cell è risolvibile in contemporanea alla soluzione dell'intero campo di moto.

I metodi fin ora citati non sempre garantiscono il rispetto della conservazione della massa nei pressi della parete. Un secondo approccio dei metodi diretti è il *cut cell approach* che a differenza degli altri è associato ad un approccio ai volumi finiti permettendo il rispetto della conservazione della massa e quantità di moto.

Qui il corpo immerso non viene discretizzato attraverso la *stairstep*, ma attraverso delle opportune tecniche le celle attraversate dal contorno vengono tagliate, scartando la parte solida, se il centro della cella è nel dominio fluido, associando parti fluide di celle con centro nel dominio solido a celle fluide limitrofe.

Ulteriori informazioni sono reperibili nella bibliografia [[4](#), [9](#), [10](#), [11](#), [14](#)]. Particolarmente interessante risulta anche l'utilizzo di metodi *Immersed Boundary* in accoppiamento con simulazioni turbolente [[6](#), [8](#), [15](#), [16](#), [18](#)].

Capitolo 3

ImmerFlow

In questo capitolo verrà presentato brevemente il solutore *ImmerFlow*. Verrà esposto basandosi sul lavoro di Gorsse et al. [20], rappresentante una primissima versione del software, ed alcune modifiche successivamente apportate da coloro che ci hanno successivamente lavorato.

Il metodo è basato su uno schema al secondo ordine ispirato ai metodi ghost-cell per flussi *incompressibili* e *compressibili*. La discretizzazione dell'equazioni avviene tramite la classica discretizzazione basata sul metodo ai *volumi finiti*.

3.1 Equazioni di Governo e Discretizzazione

Le equazioni di governo sono le *Navier-Stokes* per flussi *incomprimibili*, *inviscidi* e per *gas perfetti*:

$$\begin{aligned} \nabla \cdot \mathbf{V} &= 0 \\ \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} &= -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{V} \end{aligned} \quad 3.1$$

La discretizzazione ai *volumi finiti* prevede risoluzione delle equazioni in forma integrale, secondo la classica forma conservativa:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} \, d\Omega + \int_S \mathbf{F} \cdot \mathbf{n} \, dS = 0 \quad 3.2$$

Dove Ω è il volume rappresentante il dominio computazionale, S la sua superficie, \mathbf{W} il vettore delle variabili conservative e \mathbf{F} il vettore dei flussi convettivi:

Attraverso l'introduzione dei valori medi sul singolo volumetto ijk , la discretizzazione delle equazioni assume la seguente struttura:

$$\begin{aligned} \frac{d\mathbf{W}_{ijk}}{dt} + \frac{1}{\Delta_x} (\mathbf{F}_{i+1/2jk}^x - \mathbf{F}_{i-1/2jk}^x) + \frac{1}{\Delta_y} (\mathbf{F}_{ij+1/2k}^y - \mathbf{F}_{ij-1/2k}^y) \\ + \frac{1}{\Delta_z} (\mathbf{F}_{ijk+1/2}^z - \mathbf{F}_{ijk-1/2}^z) = 0 \end{aligned} \quad 3.3$$

dove $\mathbf{F}_{i+1/2jk}^x$ è il flusso medio nella direzione x , approssimato attraverso il attraverso il flusso numerico di Osher, ottenuto tramite la ricostruzione delle variabili conservative di tipo *MUSCL*.

L'integrazione temporale avviene attraverso lo schema al secondo ordine di Runge-Kutta:

$$\begin{aligned} \mathbf{W}^{(1)} &= \mathbf{W}^n - \Delta t \mathbf{F}(\mathbf{W}^n) \\ \mathbf{W}^{n+1} &= \mathbf{W}^n - \frac{\Delta t}{2} (\mathbf{F}(\mathbf{W}^n) + \mathbf{F}(\mathbf{W}^{(1)})) \end{aligned} \quad 3.4$$

3.2 Celle di Confine e Impermeabilità

Per le celle fluide che presentano uno o più vicini nel dominio solido viene ricostruita la condizione al contorno sull'interfaccia di separazione. La ricostruzione è realizzata in modo da rispettare la condizione di impermeabilità: la velocità normale al confine deve uguagliare quella del confine stesso, ad esempio essere nulla per corpi stazionari.

Un'accuratezza al secondo ordine è ottenuta tramite l'utilizzo della funzione *level set*, metodo ideato da Osher e Sethian [21, 22, 23]. Per un caso piano bidimensionale, una qualsiasi curva, espressa attraverso una funzione implicita, costituisce il *livello zero* da cui definire la *level set function*.

$$\begin{aligned} C(p): S^1 &\rightarrow R^2 \\ C &= \{(x, y) \mid \varphi(x, y) = 0\} \end{aligned} \quad 3.5$$

Solitamente, la *level set function* è definita come la distanza di un punto qualsiasi nel dominio dal livello zero stesso.

$$\varphi(\mathbf{x}) = \begin{cases} |\mathbf{x} - \mathbf{x}_{ib}| & \text{per punti interni al fluido} \\ -|\mathbf{x} - \mathbf{x}_{ib}| & \text{per punti interni al solido} \\ 0 & \text{sul solido} \end{cases} \quad 3.6$$

Inoltre, permette di introdurre alcune informazioni geometriche utili per la descrizione del contorno e per la discretizzazione numerica.

Tramite questa funzione è facilmente descrivibile la normale in un punto qualsiasi del dominio in cui passa la superficie iso-livello φ :

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|} \quad 3.7$$

Il metodo *level set* risulta particolarmente utile ogni qualvolta si introducano corpi in movimento rigido, ma anche superfici deformabili.

3.3 Condizioni al contorno

In questo paragrafo si tratterà brevemente come vengono imposte le condizioni al contorno nel solutore *ImmerFlow*.

In particolare, si fa riferimento alle condizioni al contorno riguardanti la velocità, essendo questo lavoro incentrato su tale grandezza. Perciò si fa riferimento alla particolare condizione al contorno nota come condizione di *no-slip wall*.

Dal punto di vista fisico, imporre la condizione al contorno equivale imporre allo strato fluido circostante il corpo la stessa velocità del corpo stesso:

- nel caso di corpo stazionario (corpo a velocità nulla investito da una corrente) equivale a imporre velocità nulla alla particella fluida adiacente;
- nel caso di corpo in movimento con ad una velocità nota equivale ad imporre, componente per componente, l'uguaglianza del vettore velocità.

A questo punto è possibile affermare che tale condiziona al contorno è una *condizione di Dirichlet* [34].

Si consideri le tre equazioni della quantità di moto. Le tre equazioni devono soddisfare globalmente le condizioni al contorno imposte dal problema, mentre non è detto che ogni singolo termine dell'equazione le rispetti, infatti, a seconda del termine analizzato, diffusivo o convettivo, questo vedrà imposta una differente condizione, ma che garantisca globalmente il soddisfacimento della condizione al contorno.

3.3.1 Termine Convettivo

Nella seguente formula viene riportato il termine convettivo per la *j-esima* componente equazioni della quantità di moto:

$$u_i \frac{\partial u_j}{\partial x_i} \quad 3.8$$

Ipotizzando di poter considerare un caso in cui sia presente solo la convezione, la condizione di *no-slip wall* si traduce in una condizione di impermeabilità che il flusso ha rispetto alla parete del solido:

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad 3.9$$

Dove con \mathbf{n} si indica la normale locale alla parete solida, mentre \mathbf{u} indica la velocità del corpo.

Il solutore *ImmerFlow* impone esattamente questa condizione, per poi ricostruire il valore di velocità all'interfaccia *staircase* di separazione cella solida e fluida.

La velocità di una cella fluida limitrofa alla *staircase* viene proiettata sul corpo, dove viene imposta la condizione dell'equazione 3.9: mantenendo la componente tangenziale del flusso, ma sostituendo la componente normale con quella del corpo stesso.

L'ultimo passaggio è ricostruire la velocità dalla superficie solida al centroide dell'interfaccia della *staircase*.

3.3.2 Termine Diffusivo

Il contributo diffusivo è:

$$\nu \frac{\partial^2 u_j}{\partial x_i \partial x_i} \quad 3.10$$

Essendo il flusso viscoso, presentante viscosità dinamica e viscosità cinematica nulle, nascono degli sforzi viscosi che portano la velocità esterna al fluido ad uguagliare la velocità imposta dal corpo.

$$\mathbf{u} = \mathbf{u}_B \quad 3.11$$

Come qui sopra esposto, il software *ImmerFlow* applica esattamente queste condizioni.

Anche in questo caso, l'ultimo passaggio è la ricostruzione della velocità, componente per componente, sull'interfaccia della *staircase*.

Capitolo 4

I metodi di Interpolazione

In questo capitolo viene riportata l'articolo su cui si basa il lavoro di questa tesi (Kang, Iaccarino e Moin et al. [3]). Brevemente, si presenteranno vari approcci di interpolazione polinomiale a differenti gradi in sostituzione all'equazioni del moto all'interfaccia virtuale, quella precedentemente indicata con *staircase boundary*, di separazione solido-fluido. Lo scopo di questa interpolazione è la semplificazione del calcolo del campo di velocità, le quali necessitano solo di conoscere il campo di velocità in alcune celle limitrofe e del valore della condizione al contorno a parete.

4.1 Origini

In generale, i solutori *Immersed Boundary* che si basano su un approccio di interpolazione si contrappongono a tutti quei metodi che cercano di imporre le condizioni al contorno attraverso l'introduzione di forzanti nell'equazione della quantità di moto. Questi metodi, di cui precedentemente se ne può visualizzare alcuni esempi, prevedono l'introduzione di forze per unità di volume presentanti, talvolta, alcune costanti da determinare *ad hoc* per ciascun problema a seconda della frequenza del flusso, riducendo la solidità del codice. In aggiunta, all'aumentare del valore delle costanti le equazioni diventano sempre più rigide da risolvere, implicando una notevole riduzione dell'intervallo di integrazione per flussi non stazionari e di conseguenza l'aumento del costo computazionale. Degno di nota è comunque il fatto che i metodi *Immersed Boundary* con l'approccio del forcing rimangono *più convenienti* dei metodi *Body Fitted*.

L'idea di base dell'approccio di interpolazione è esposta in Fadlun [1]. In questo lavoro vengono riportate tre diversi schemi di interpolazione.

Il più semplice, prevede nell'assegnare come soluzione all'interfaccia il valore stesso della condizione al contorno. In questo caso l'interpolazione è di grado zero, assumendo la velocità del contorno costante su tutta la cella.

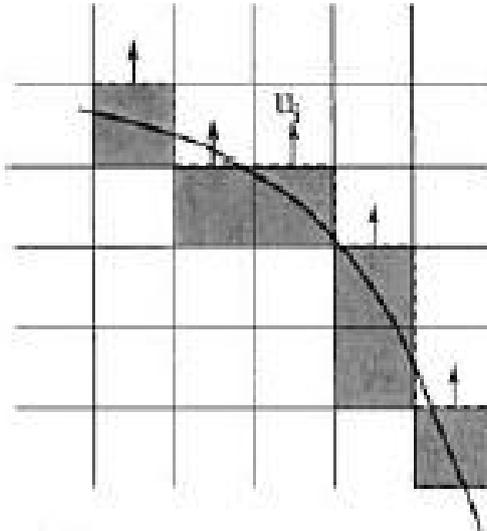


Figura 7: interpolazione a grado zero [1].

Un semplice evoluzione, prevede che si assegni il valore della condizione al contorno pesata. Come peso viene proposto la frazione volumica di cella occupata del solido: Ω_b/Ω_{TOT} .

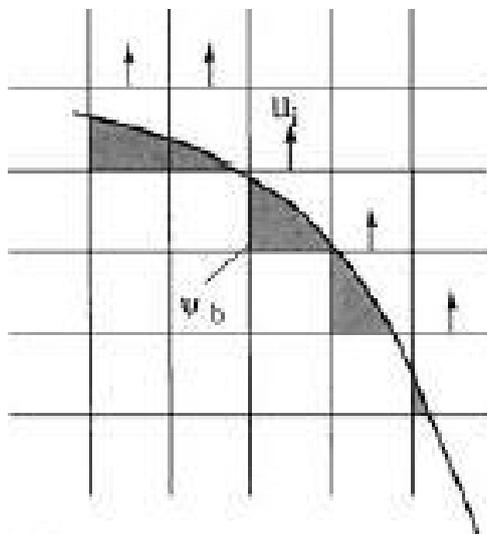


Figura 8: interpolazione pesata [1].

Il terzo, ed ultimo approccio proposto, prevede l'approssimazione della velocità con una interpolazione lineare, compresa tra il valore a parete ed il valore della cella fluida più vicina.

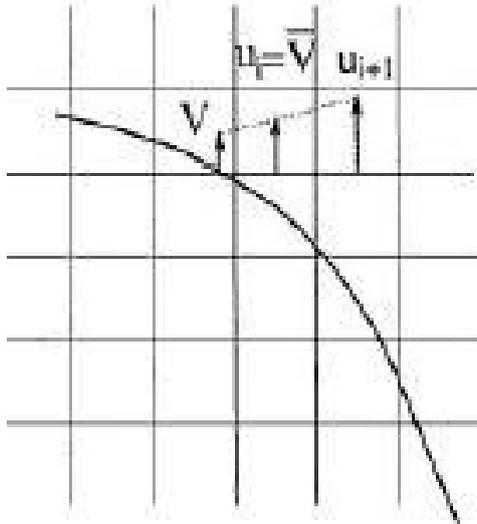


Figura 9: interpolazione lineare [1].

4.2 Descrizione del Solutore

Le equazioni di governo del problema sono le *Navier-Stokes* per flussi non stazionari e incompressibili, in forma adimensionale.

$$\begin{cases} \nabla \cdot \mathbf{V} = 0 \\ \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{V} \end{cases} \quad 4.1$$

In forma discreta, la soluzione è ricavata da un metodo *Crank-Nicolson* per il termine diffusivo e da un metodo *Runge-Kutta* al terzo ordine per i termini convettivi, ottenendo per la k -esima iterazione un'equazione della forma

$$\begin{aligned} \frac{u_i^k - u_i^{k-1}}{\Delta t} + \alpha_k \left(\frac{\partial u_i u_j}{\partial x_j} \right)^{k-1} + \beta_k \left(\frac{\partial u_i u_j}{\partial x_j} \right)^{k-2} \\ = -(\alpha_k + \beta_k) \frac{\partial p^{k-1}}{\partial x_i} + \frac{\beta_k + \alpha_k}{2 Re} \left[\frac{\partial^2 u_i^k}{\partial x_j^2} + \frac{\partial^2 u_i^{k-1}}{\partial x_j \partial x_j} \right] + O(\Delta t^2) \end{aligned} \quad 4.2$$

Da cui, con l'utilizzo di un metodo denominato *fractional-step*, si ottiene preventivamente una prima soluzione \widehat{u}_i^k che non rispetta la condizione di solenoidalità del campo, e successivamente si introduce una correzione sia al termine di pressione che alla velocità per imporre questo vincolo.

La correzione si ottiene risolvendo un sistema lineare su tutte le celle, nella forma:

$$\frac{\partial^2 \Phi}{\partial x_j \partial x_j} = \frac{1}{\Delta t} \frac{1}{\alpha_k + \beta_k} \frac{\partial \widehat{u}_i^k}{\partial x_i} \quad 4.3$$

Da cui si possono imporre le seguenti correzioni:

$$u_i^k = \widehat{u}_i^k - (\alpha_k + \beta_k) \Delta t \frac{\partial \Phi}{\partial x_i} \quad p^k = p^{k-1} + \Phi \quad 4.4$$

4.3 Schemi di Interpolazione

Nel seguito di questo capitolo si indicherà con la sigla *IB* un punto sulla superficie di un corpo giacente sulla normale, al corpo stesso, passante un'interfaccia *stairstep*.

4.3.1 Interpolazione Lineare o Standard Reconstruction Method

Si consideri un sistema di riferimento cartesiano centrato in un punto IB . Si considerino anche due punti fluidi limitrofi, 1 e 2. Questi tre punti devono essere scelti in modo da costruire un triangolo intorno all'interfaccia in analisi, come riportato nella figura 13.

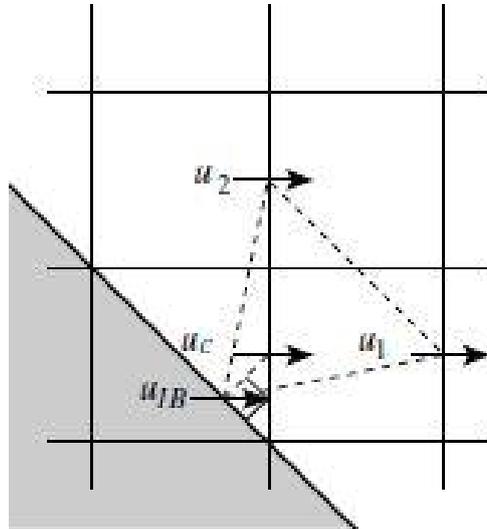


Figura 10: stencil lineare [3].

L'interpolazione per una delle componenti di velocità può avvenire nella seguente forma:

$$u_{i,c}^k = \omega_1 u_{i,1}^k + \omega_2 u_{i,2}^k + \omega_{IB} u_{i,IB}^k \quad 4.5$$

Oppure, sfruttando il sistema di riferimento cartesiano, precedentemente introdotto, nella forma

$$u_i^k = a_1 x_1 + a_2 x_2 + u_{i,IB}^k \quad 4.6$$

In quest'ultimo caso, i coefficienti di interpolazione possono essere calcolati risolvendo un semplice sistema lineare, una volta nota la soluzione sostituendo le coordinate del punto C, si ottiene la velocità interpolata del nodo di interfaccia.

Attraverso un'analisi con espansioni in serie di Taylor, è dimostrabile che questa interpolazione presenta un'accuratezza al secondo ordine in spazio ed un'accuratezza al primo ordine in tempo. Inoltre, lo schema non presenta grandi limitazioni sull'intervallo di integrazione, anzi il vincolo non è imposto dalle limitazioni delle celle circostanti.

Nel caso di utilizzo del metodo *fractional-step*, l'interpolazione può essere svolta sul primo step dell'integrazione, ricavando la velocità $\widehat{u}_{i,c}^k$.

4.3.1.1 Problematiche evidenziate

L'approccio non prende in considerazione il campo di pressione nell'intorno del punto IB : non c'è contributo esplicito del campo di velocità al precedente istante, come anche per il gradiente di pressione. Questo ha evidenziato forti fluttuazioni del campo di pressione intorno a IB .

A causa del disaccoppiamento tra i campi di velocità e di pressione nell'intorno di IB , diventa complicato soddisfare la condizione di divergenza nulla nel volume di controllo in cui è contenuto.

Considerando ad esempio il volume di controllo della figura 14.

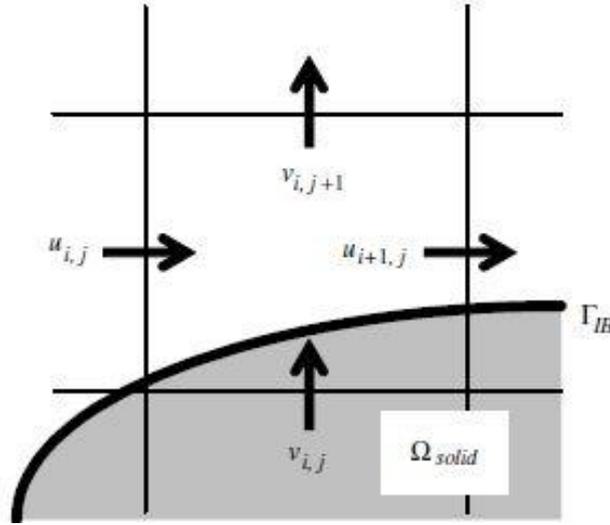


Figura 11: volume di controllo [3].

Lo schema di interpolazione interverrebbe per determinare le velocità indicate in figura con $u_{i,j}$, $u_{i+1,j}$ e $v_{i,j+1}$: per la velocità $v_{i,j}$ se si imponesse valore nullo, sarebbe impossibile soddisfare l'equazione di continuità sul volume di controllo. Al contrario, se si imponesse un valore non nullo, determinato dalla equazione della quantità di moto, la soluzione perderebbe senso fisico nel campo di pressione.

Ultima problematica evidenziata, riguarda l'utilizzo dell'interpolazione lineare con il metodo *fractional-step*: è osservato un continuo incremento di pressione nel volume di controllo dei punti IB , dovuto principalmente ad un valore finito di $\nabla \cdot \hat{\mathbf{u}}$.

4.3.2 Interpolazione Lineare con Gradiente di Pressione

Per far fronte alle problematiche di disaccoppiamento dei campi di pressione e velocità, si può pensare di riscrivere l'equazione di interpolazione. Si può allora intervenire sul delta di velocità espresso come la differenza tra la componente di velocità predetta e la componente completa dell'istante precedente:

$$\Delta u_i^k = \widehat{u}_i^k - u_i^{k-1} \quad 4.7$$

La formula di interpolazione assume la stessa forma dell'[equazione 4.4](#),

$$\Delta u_{i,c}^k = \omega_1 \Delta u_{i,1}^k + \omega_2 \Delta u_{i,2}^k + \omega_{IB} \Delta u_{i,IB}^k \quad 4.8$$

È possibile ricavare, una formulazione che comprenda sia $\widehat{u}_{i,c}^k$, sia $\Delta u_{i,c}^k$:

$$\begin{aligned} (1 - \eta) \widehat{u}_{i,c}^k - \eta \Delta u_{i,c}^k \\ = (1 - \eta)(\omega_1 u_{i,1}^k + \omega_2 u_{i,2}^k + \omega_{IB} u_{i,IB}^k) + \eta(\omega_1 \Delta u_{i,1}^k + \omega_2 \Delta u_{i,2}^k \\ + \omega_{IB} \Delta u_{i,IB}^k) \end{aligned} \quad 4.9$$

Per un caso bidimensionale come quello di figura 14, si ottiene:

$$\widehat{u}_{i,c}^k = \omega_1 u_{i,1}^k + \omega_2 u_{i,2}^k + \omega_{IB} u_{i,IB}^k + \eta(u_{i,c}^{k-1} - \omega_1 u_{i,1}^k - \omega_2 u_{i,2}^k - \omega_{IB} u_{i,IB}^k) \quad 4.10$$

Con $\eta = \sqrt{(\omega_1 + \omega_2)/\omega_{IB}}$.

In aggiunta, per ridurre il disaccoppiamento tra velocità e gradiente di pressione, si possono aggiungere termini

$$\widehat{u}_{i,c}^k = \omega_{i,1} u_{i,1}^k + \omega_{i,2} u_{i,2}^k + \omega_{i,IB} u_{i,IB}^k + \eta (u_{i,c}^{k-1} - \omega_1 u_{i,1}^{k-1} - \omega_2 u_{i,2}^{k-1} - \omega_{IB} u_{i,IB}^{k-1}) - \delta_k \Delta t \left(\left. \frac{\partial p^{k-1}}{\partial x_i} \right|_c - \omega_{i,1} \left. \frac{\partial p^{k-1}}{\partial x_i} \right|_1 - \omega_{i,2} \left. \frac{\partial p^{k-1}}{\partial x_i} \right|_2 - \omega_{i,IB} \left. \frac{\partial p^{k-1}}{\partial x_i} \right|_{IB} \right) \quad 4.11$$

Come lo schema originario anche questo mantiene un buon livello di accuratezza: al secondo ordine in spazio e al primo ordine in tempo. Parziale degradazione della precisione può essere introdotta dal calcolo del gradiente di pressione in IB, mentre gli altri termini sono noti già dalla risoluzione del campo nei nodi fluidi. Un altro punto a sfavore presentato da questo schema è l'introduzione di parametri, $\omega_{i,j}$, che devono essere introdotti da utente oppure dovranno essere calcolati a seconda della geometria

4.3.3 Interpolazione Quadratica

Considerando un sistema di riferimento cartesiano locale, centrato nel nodo di interfaccia C , una componente di velocità in un'approssimazione quadratica con la seguente formula:

$$u_i^k = a_1 x_1^2 + a_2 x_2^2 + b_1 x_1 + b_2 x_2 + u_{i,c}^k \quad 4.12$$

Le incognite per poter esprimere la velocità sono cinque, mentre si può ricorrere a due metodi alternativi per il loro calcolo.

Il primo, simile a quanto accade per lo schema lineare, consiste nello scegliere cinque punti circostanti in cui la soluzione è nota, risolvere un semplice sistema algebrico lineare, con cui poi calcolare la velocità incognita. Così facendo però si ripresenterebbe il problema di disaccoppiamento tra velocità e campo di pressione, precedentemente introdotto.

Da cui il secondo approccio, vengono individuati solamente quattro nodi vicini, come quinta equazione, per chiudere il sistema algebrico lineare si impone il rispetto della equazione discreta della quantità di moto: sostituendo l'[equazione 4.10](#) nell'[equazione 4.1](#), ottenendo:

$$\begin{aligned} \frac{\widehat{u}_{i,c}^k}{\Delta t} - \frac{\alpha_k + \beta_k}{\text{Re}} (a_{i,1}^k + a_{i,2}^k) \\ = \frac{u_{i,c}^{k-1}}{\Delta t} - (\alpha_k + \beta_k) \frac{\partial p^{k-1}}{\partial x_i} - \alpha^k \left(\frac{\partial u_i u_j}{\partial x_j} \right)^{k-1} - \beta_k \left(\frac{\partial u_i u_j}{\partial x_j} \right)^{k-2} \\ + \frac{\alpha_k + \beta_k}{2 \text{Re}} \frac{\partial^2 u_i^{k-1}}{\partial x_j \partial x_j} + O(\Delta t^2) \end{aligned} \quad 4.13$$

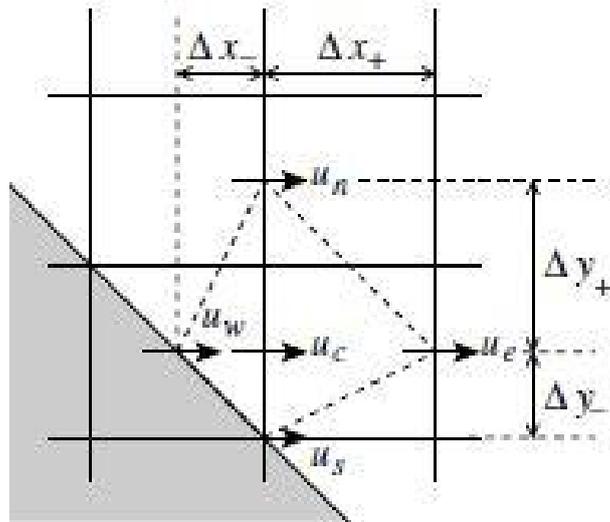


Figura 12: quadratic stencil [3].

Ancora una volta, l'accuratezza spaziale mantiene il secondo ordine, mentre quella temporale rimane sempre al primo.

Capitolo 5

Schemi di Interpolazione

In questo capitolo, verranno presentati gli approcci di interpolazione utilizzati durante lo svolgimento di questo lavoro di tesi magistrale. Rispetto a quelli presentati da Iaccarino et al [3], viene introdotto anche uno schema di interpolazione lineare lungo la retta normale alla superficie del corpo. L'interpolazione si sostituisce alla soluzione delle *Navier-Stokes* ad un'interfaccia di separazione solido-fluido, una volta nota la soluzione anche all'interfaccia sarà possibile avanzare all'iterazione $k+1$ -esima. A questo punto si ripeterà il processo risolvendo prima il campo fluido e poi nuovamente ricostruendo la soluzione per interpolazione.

Il codice è stato scritto in linguaggio C++ [29, 30]) con il supplemento di git [31].

5.1 Interpolazione Lineare

Il primo schema di interpolazione implementato in questa tesi consiste in una semplice interpolazione lineare. L'interpolazione avviene lungo la normale al corpo passante per il centroide dell'interfaccia analizzata.

Si consideri un'interfaccia *staircase* di separazione fluido-solido, vengono definiti i seguenti punti utilizzati anche per gli schemi successivi:

1. Punto di interfaccia, o *interface point*, indicato con la sigla **IP**.
2. Punto di proiezione, o *projection point*, indicato con la sigla **PP**: definito dall'intersezione tra la superficie del corpo immerso e la normale al corpo stesso passante per il centroide del punto **IP**.
3. Punto normale, o *normal point*, indicato con la sigla **NP**: un punto nella cella fluida limitrofa all'interfaccia definito in modo che la distanza, lungo la normale, tra i punti **IP** e **PP** sia pari alla distanza tra i punti **IP** e **NP**.

L'obiettivo finale è il calcolo delle tre componenti di velocità per ogni punto **IP** presente nel dominio computazionale. Risulta quindi necessario conoscere due valori nell'intorno di tale punto e i candidati idonei sono esattamente i rimanenti due punti sopra definiti. Nella figura seguente, figura 16, si possono vedere questi come sono posizionati questi tre punti per una generica configurazione geometrica bidimensionale.

Nella figura 16 si può vedere tratteggiata in grigio la griglia cartesiana, in nero con una linea continua spessa è riportato il corpo immerso in tale griglia, la normale al corpo è indicata in nero con una linea tratteggiata nera, sempre in nero con una linea sottile spezzata è riportata il contorno virtuale, infine in rosso sono indicati: **PP** rappresentato con +, **NP** visualizzabile con un quadratino e, infine, **IP** mostrato con ×.

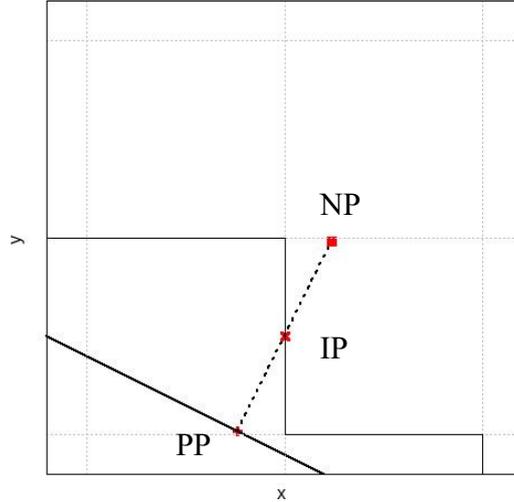


Figura 13: stencil di interpolazione lineare.

Per quanto riguarda i termini convettivi dell'equazione di *Navier-Stokes*, tramite un polinomio è possibile ricostruire la velocità di **NP** partendo dalla soluzione della cella fluida che lo contiene. Attraverso un polinomio di grado zero si assegnerebbe la velocità della centroide della cella, aumentando il grado di interpolazione si potrebbe estrapolare con maggiore accuratezza il valore di ogni singola componente di velocità. Nel corso di questo elaborato è stato preso in considerazione un polinomio di *grado zero*.

Nota la condizione al contorno imposta dal solido è ora possibile ricavare il valore della velocità all'interfaccia. Indicando con d_{IP} la distanza tra **PP** e **IP**, con d_{PN} la distanza tra **PP** e **NP**, il valore della *i-esima* componente di velocità è calcolabile, con l'[equazione 5.1](#):

$$u_i^{IP} = \frac{d_{IP}}{d_{PN}} u_i^{NP} + \left(1 - \frac{d_{IP}}{d_{PN}}\right) u_i^{PP} \quad 5.1$$

Per quanto riguarda la parte diffusiva dell'equazione, in questo caso non viene implementato nessuno schema, in quanto il codice originario di *ImmerFlow* presenta un algoritmo simile per calcolare tale contributo.

5.2 Interpolazione Multi Lineare

In questo caso l'interpolazione prende spunto da quanto presentato da Iaccarino (et al. [3]), riportato in questo testo al [paragrafo 4.3.1](#) del capitolo 4.

5.2.1 Convezione

Lo schema di interpolazione per i contributi convettivi punta a ricostruire i valori delle componenti di velocità da cui è poi possibile calcolare i valori dei flussi. È stato implementato l'interpolazione riportata nell'[equazione 4.4](#), mentre lo stencil di interpolazione è riportato in figura 17.

$$u_{i,IP}^k = \omega_1 u_{i,1}^k + \omega_2 u_{i,2}^k + \omega_{PP} u_{i,PP}^k \quad 5.2$$

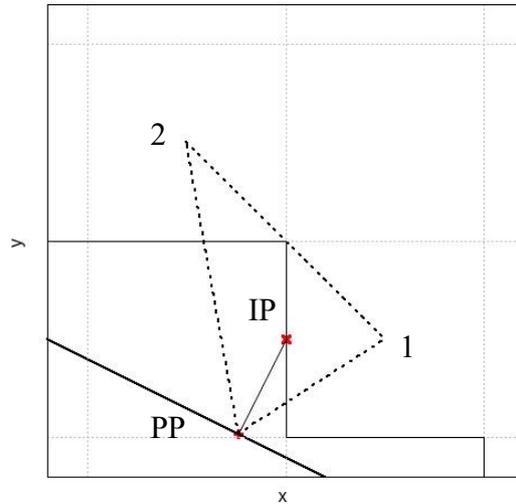


Figura 14: stencil di interpolazione multi lineare.

Note le componenti di velocità nei vertici del triangolo di interpolazione, il calcolo della velocità avviene a seguito della scelta dei pesi di interpolazione necessari.

5.2.1.1 Coefficienti di Interpolazione

Bisogna fin da subito evidenziare che purtroppo non è stato trovato un singolo algoritmo che ricopra sia casi bidimensionali sia casi tridimensionali, anche se il concetto alla base è simile.

In entrambi i casi il primo passo è l'individuazione, rispettivamente, di due e tre vicini che ricadano nel dominio fluido al di fuori dell'*interfaccia virtuale* e verificare che il nodo sull'interfaccia ricada all'interno del triangolo.

5.2.1.1.1 Caso Bidimensionale

Il triangolo rappresentante lo stencil di interpolazione è suddiviso in tre triangoli a partire dai tre vertici **1**, **2** e **PP**, come riportato in figura 18. Il peso che ogni vertice ha sull'interpolazione è pari al valore dell'area del triangolo opposto rapportato al valore dell'area totale del triangolo originario, per i tre vertici si ha:

$$\begin{aligned} \omega_1 &= \frac{A_1}{A} \\ \omega_2 &= \frac{A_2}{A} \\ \omega_{PP} &= \frac{A_{PP}}{A} \end{aligned} \tag{5.3}$$

Il calcolo di tali grandezze geometriche è molto semplice, infatti, note le posizioni di tutti i quattro punti di figura 18, il calcolo delle aree è semplicemente la metà della norma in base due del prodotto vettore di due vettori descrittivi il triangolo.

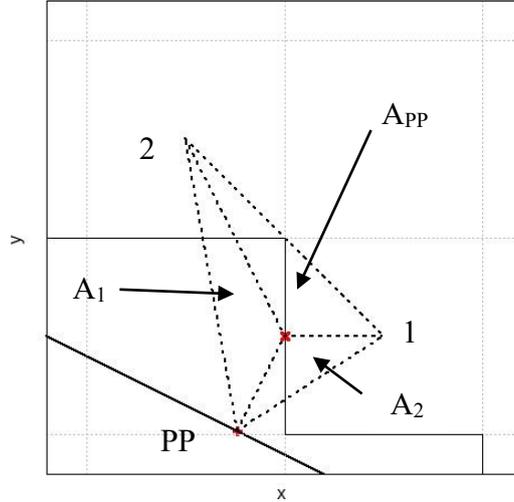


Figura 15: rappresentazione delle aree interne allo stencil.

Si riporta qui di seguito un caso rappresentativo per il calcolo del peso del punto **PP**. Considerando i vettori che da **IP** punta verso i nodi 1 e 2, rispettivamente $\overrightarrow{IP1}$ e $\overrightarrow{IP2}$, l'area A_{PP} risulta:

$$A_{PP} = \frac{1}{2} \|\overrightarrow{IP1} \times \overrightarrow{IP2}\| \quad 5.4$$

Mentre considerando i vettori $\overrightarrow{PP1}$ e $\overrightarrow{PP2}$ con una formulazione omologa è possibile calcolare l'area del triangolo, oppure semplicemente note le tre aree sommandone i contributi. Dal punto di vista del calcolo numerico, converrebbe il secondo metodo in quanto il prodotto vettore ha un elevato costo computazionale rispetto alla somma. A questo punto, il peso per il nodo **PP** è fornito dalla [equazione 5.3](#).

5.2.1.1.2 Caso Tridimensionale

In questo caso, lo stencil di interpolazione è visualizzabile come una piramide, al cui interno si presenta il punto IP. I pesi vengono calcolati in modo simile al caso bidimensionale, solo sostituendo all'area di un triangolo il volume di una delle quattro piramidi e all'area totale del triangolo il volume totale della piramide. Si ricorda che il volume di una piramide definita da tre vettori condividenti il polo d'origine è un terzo del valore assoluto del prodotto misto dei tre vettori.

$$V = \frac{1}{3} |(\overrightarrow{V_0V_1} \times \overrightarrow{V_0V_2}) \cdot \overrightarrow{V_0V_3}| \quad 5.5$$

Dove $\overrightarrow{V_0V_1}$, $\overrightarrow{V_0V_2}$ e $\overrightarrow{V_0V_3}$ sono i tre vettori contenuti tra i quattro vertici della piramide V_0, V_1, V_2, V_3 .

5.2.2 Diffusione

Per i contributi diffusivi ciò che conta sono i gradienti delle componenti di velocità, in questo paragrafo si riporta come questo è stato implementato. Considerando la figura 18 e l'[equazione 5.2](#) si osserva che, per un fissato istante temporale, lo stencil di integrazione è fissato, ovvero le posizioni dei nodi sono fisse, come sono anche costanti i valori delle velocità nei nodi fluidi. Risulta evidente come il gradiente di una generica componente di velocità in **IP** sia esprimibile [[24](#), [25](#)]:

$$\frac{\partial u_{i,IP}^k}{\partial x_j} = \frac{\partial \omega_1}{\partial x_j} u_{i,1}^k + \frac{\partial \omega_2}{\partial x_j} u_{i,2}^k + \frac{\partial \omega_{PP}}{\partial x_j} u_{i,PP}^k \quad 5.6$$

Per poter svolgere il calcolo resta solo da capire come variano i pesi di interpolazione se il nodo **IP** avesse uno spostamento infinitesimo nelle direzioni cartesiane.

5.2.2.1 Coefficienti di Interpolazione

5.2.2.1.1 Caso Bidimensionale

Per il caso bidimensionale sono sufficienti calcolare le componenti del gradiente solo nelle direzioni dell'asse x e dell'asse y.

In generale, vale la seguente uguaglianza:

$$\frac{\partial \omega_i}{\partial x_j} = \frac{1}{A} \frac{\partial A_i}{\partial x_j} \quad 5.7$$

Senza perdere di generalità, il calcolo delle derivate dei pesi di interpolazione si riducono al semplice calcolo delle derivate delle aree dei triangoli di figura 18. Considerando un triangolo qualsiasi, figura 19, bisogna calcolare la variazione dell'area da esso racchiusa al variare della posizione di uno dei vertici.

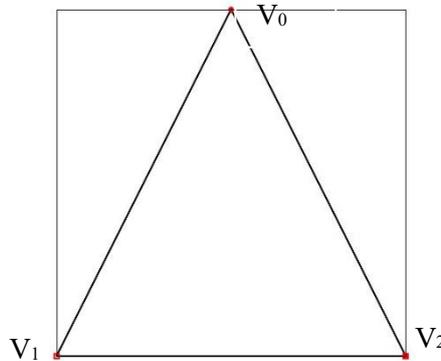


Figura 16: un triangolo generico.

Esprimendo, il vettore normale al triangolo come il prodotto vettore di due vettori con origine nel vertice V_0 e fine nei vertici V_1 e V_2 :

$$\vec{n} = \overrightarrow{V_0V_1} \times \overrightarrow{V_0V_2} \quad 5.8$$

L'area del triangolo risulta:

$$A = \frac{1}{2} \sqrt{\vec{n} \cdot \vec{n}} \quad 5.9$$

E la sua derivata:

$$\frac{\partial A}{\partial x_j} = \frac{1}{2} \frac{1}{\sqrt{\vec{n} \cdot \vec{n}}} \vec{n} \cdot \frac{\partial \vec{n}}{\partial x_j} \quad 5.10$$

Bisogna quindi calcolare le derivate del vettore normale nelle direzioni cartesiane, partendo dalla derivata lungo l'asse x:

$$\begin{aligned} \frac{\partial \vec{n}}{\partial x} &= \frac{\partial \overrightarrow{V_0V_1} \times \overrightarrow{V_0V_2}}{\partial x} = \frac{\partial \overrightarrow{V_0V_1}}{\partial x} \times \overrightarrow{V_0V_2} + \overrightarrow{V_0V_1} \times \frac{\partial \overrightarrow{V_0V_2}}{\partial x} \\ &= \frac{\partial(V_1 - V_0)}{\partial x} \times \overrightarrow{V_0V_2} + \overrightarrow{V_0V_1} \times \frac{\partial(V_2 - V_0)}{\partial x} \end{aligned} \quad 5.11$$

Considerando come vertice “mobile” il vertice V_0 e definendo il versore dell’asse x: $\hat{i} = (1,0,0)^T$:

$$\begin{aligned} \frac{\partial \vec{n}}{\partial x} &= -\frac{\partial V_0}{\partial x} \times \overrightarrow{V_0V_2} - \overrightarrow{V_0V_1} \times \frac{\partial V_0}{\partial x} = -\hat{i} \times \overrightarrow{V_0V_2} - \overrightarrow{V_0V_1} \times \hat{i} = (\overrightarrow{V_0V_2} - \overrightarrow{V_0V_1}) \times \hat{i} \\ &= (\overrightarrow{V_2} - \overrightarrow{V_1}) \times \hat{i} = \overrightarrow{\Delta V} \times \hat{i} = -\Delta V_y \hat{k} = (y_2 - y_1) \hat{k} \end{aligned} \quad 5.12$$

Allo stesso modo si ricava la derivata in y, sostituendo al versore \hat{i} il versore dell’asse y $\hat{j} = (0,1,0)^T$:

$$\frac{\partial \vec{n}}{\partial y} = (x_1 - x_2) \hat{k} \quad 5.13$$

Da notare che, definiti i vertici del triangolo V_1 e V_2 , le derivate siano indipendenti dalla posizione del vertice mobile V_0 .

Riconsiderando la figura 18, si potrebbe verificare che il nodo di interfaccia risulti appartenente ad uno dei lati del triangolo, portando ad alcuni problemi di stabilità dell’algoritmo, perciò data l’indipendenza delle derivate dei pesi dal nodo dalla posizione del nodo di interfaccia, si è scelto di calcolare le derivate con nodi di riferimento il centroide del triangolo (figura 18):

$$V_C = \frac{1}{3}(V_1 + V_2 + V_{PP}) \quad 5.14$$

5.2.2.1.2 Caso Tridimensionale

Il caso tridimensionale risulta decisamente più semplice. In questo caso vale la seguente uguaglianza:

$$\frac{\partial \omega_i}{\partial x_j} = \frac{1}{V} \frac{\partial V_i}{\partial x_j} \quad 5.15$$

Il volume di una generica piramide, figura 20, di vertici V_1, V_2, V_3 e V_0 , mentre V_C il centroide dei vertici della base compresa tra V_1, V_2, V_3 :

$$V = \frac{1}{3} A \hat{n} \cdot \overrightarrow{V_0V_C} \quad 5.15$$

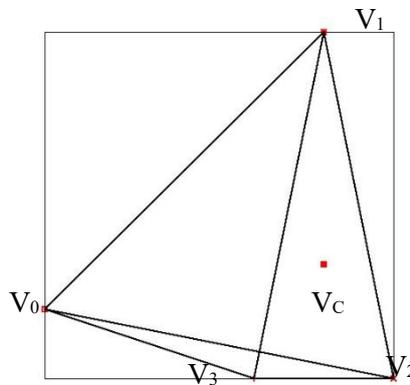


Figura 17: piramide generica.

Dove \hat{n} è il versore della base della piramide e $\overrightarrow{V_0V_C}$ il vettore che punta da V_0 a V_C . il vertice V_0 è l’unico a poter variare la sua posizione, perciò:

$$\frac{\partial V}{\partial x_j} = \frac{1}{3} A \hat{n} \cdot \frac{\partial \overline{V_0 V_C}}{\partial x_j} = \frac{1}{3} A \hat{n} \cdot \hat{e}_j = \frac{1}{3} A \frac{\vec{n}}{\|\vec{n}\|_2} \cdot \hat{e}_j = \frac{1}{3} \frac{\|\vec{n}\|_2}{2} \frac{\vec{n}}{\|\vec{n}\|_2} \cdot \hat{e}_j = \frac{1}{6} n_i \quad 5.16$$

Dove \vec{n} è il prodotto vettore di due vettori costituenti la base della piramide, $\vec{n} = \overline{V_3 V_1} \times \overline{V_3 V_2}$.

5.3 Interpolazione Multi Lineare con Gradiente di Pressione

Partendo dall'idea presentata al [paragrafo 4.3.2](#), si introduce allo schema multilineare presentato al [paragrafo 5.2](#) termine aggiuntivo che include il gradiente di pressione in alcuni nodi:

$$u_{i,IP}^k = \omega_{i,1} u_{i,1}^k + \omega_{i,2} u_{i,2}^k + \omega_{i,IB} u_{i,PP}^k - \delta_k \frac{\Delta t}{\rho} \left(\frac{\partial p^{k-1}}{\partial x_i} \Big|_{IP} - \omega_{i,1} \frac{\partial p^{k-1}}{\partial x_i} \Big|_1 - \omega_{i,2} \frac{\partial p^{k-1}}{\partial x_i} \Big|_2 - \omega_{i,IB} \frac{\partial p^{k-1}}{\partial x_i} \Big|_{PP} \right) \quad 5.17$$

Per quanto riguarda il calcolo dei coefficienti sia per il caso convettivo sia per il caso diffusivo si faccia riferimento, rispettivamente, al [paragrafo 5.2.1](#) e al [paragrafo 5.2.2](#).

In questo capitolo ci si concentra sul calcolo dell'unica incognita rimanente per poter chiudere lo schema di interpolazione: il gradiente di pressione a parete in **PP**.

Considerando una terna versore costituita da due versori tangenziali ed una normale alla superficie del corpo immerso \hat{t}_1 , \hat{t}_2 e \hat{n} , vale la seguente relazione:

$$\frac{\partial p}{\partial x_j} = \frac{\partial p}{\partial n} n_j + \frac{\partial p}{\partial t_1} t_{1,j} + \frac{\partial p}{\partial t_2} t_{2,j} \quad 5.18$$

Nell'approssimazione di strato limite laminare, si può considerare il gradiente di pressione normale a parete nullo.

Il primo versore tangenziale è approssimato a parallelo alla direzione alla velocità nella cella fluida più vicina all'interfaccia, ad esempio la cella indicata con 1 in figura 17, e successivamente normalizzato a modulo unitario. Il secondo è scelto come il prodotto vettoriale tra la normale \hat{n} e il versore \hat{t}_1 . Il valore delle derivate nelle direzioni tangenziali a parete è affidato a funzioni già incluse nel software *ImmerFlow*.

5.4 Interpolazione Quadratica

Lo schema di interpolazione quadratica presenta una approssimazione con un polinomio parabolico in tutte le direzioni di interpolazione, per il calcolo dei coefficienti è stato necessario risolvere un piccolo sistema lineare partendo dal campo di velocità delle vicine celle fluide.

5.4.1 Convezione

Si consideri un sistema di riferimento cartesiano centrato in **PP**, si approssimi la velocità in prossimità di tale punto con un polinomio del secondo ordine:

$$u_i^k = a_1 x^2 + a_2 y^2 + a_3 z^2 + b_1 x + b_2 y + b_3 z + u_{i,PP}^k \quad 5.18$$

Servono sei celle fluide per poter calcolare i coefficienti in un caso tridimensionale, quattro per un caso bidimensionale. Per trovarle, è stato implementato un algoritmo che permettesse di trovare le celle fluide più vicine all'interfaccia d'interesse. L'algoritmo, per un caso bidimensionale, porta ad avere stencil di interpolazione del tipo in figura 21:

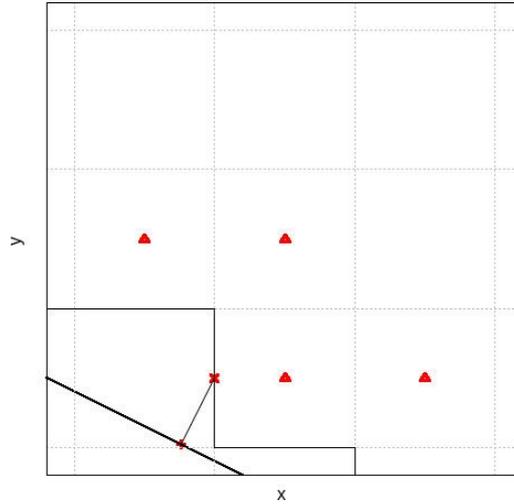


Figura 18: stencil di interpolazione parabolica.

Per ogni interfaccia, per ogni componente di velocità è possibile scrivere un sistema lineare avente come incognite i coefficienti, come termine noto le componenti di velocità delle celle riferite alla velocità di **PP**:

$$\begin{bmatrix} x_1^2 & y_1^2 & x_1 & y_1 \\ x_2^2 & y_2^2 & x_2 & y_2 \\ x_3^2 & y_3^2 & x_3 & y_3 \\ x_4^2 & y_4^2 & x_4 & y_4 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{Bmatrix} = \begin{Bmatrix} u_{i,1}^k \\ u_{i,2}^k \\ u_{i,3}^k \\ u_{i,4}^k \end{Bmatrix} - u_{i,PP}^k \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} \quad 5.19$$

Ottenuta la soluzione del sistema, sostituendo all'[equazione 5.18](#) la posizione di **IP**, si ricava la *i*-esima componente di velocità.

Il sistema va risolto tante volte quante sono le dimensioni del problema. Da notare che gli elementi della matrice sono invarianti, in quanto contengono solo informazioni geometriche della griglia, ma nonostante questo i coefficienti variano a seconda della componente studiata.

5.4.1.1 Casi Particolari

Particolare attenzione richiedono quei punti di interfaccia che presentano la normale al corpo immerso pressoché allineata con la normale dell'interfaccia. Questi casi presenterebbero lo stencil di figura 22.

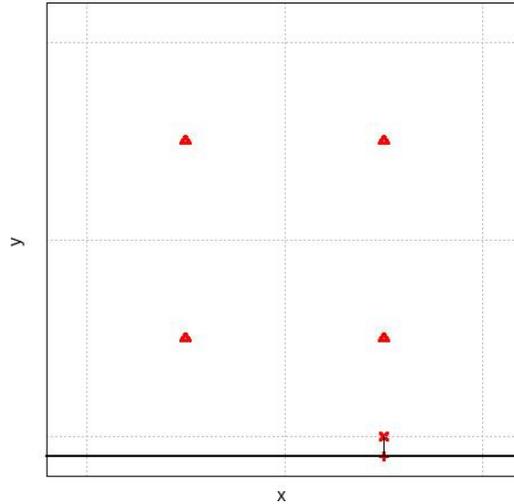


Figura 19: stencil in un caso particolare.

È stato osservato che tali configurazioni presentassero matrici del sistema mal condizionate [34], rendendo impossibile la risoluzione del sistema lineare. Questo problema è stato risolto riducendo di una dimensione l'interpolazione, mantenendo un polinomio di secondo grado, ed interpolando lungo la normale al corpo. Ad esempio, se il problema fosse bidimensionale, si svolgerebbe un'interpolazione monodimensionale.

Il procedimento per svolgere questa semplificazione del sistema è semplice. Prima di tutto si individua il punto lungo la normale all'interfaccia appartenente al corpo, denominato **UP**, successivamente si rimuovono le celle il cui centro non giace sulla direttrice della normale all'interfaccia. Per il caso in figura 22, il punto **UP** coincide con il punto **IP**. Lo stencil finale può essere visualizzato in figura 23.

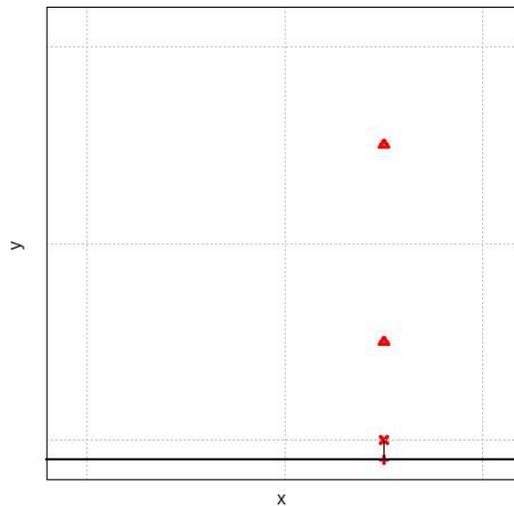


Figura 20: stencil modificato per interpolazione parabolica.

5.4.2 Diffusione

In questo caso, essendo l'interpolazione svolta con un polinomio, il calcolo delle derivate risulta molto semplificato, infatti:

$$\frac{\partial u_i^k}{\partial x_j} = 2a_j x_j + b_j \quad 5.20$$

Una volta noti i coefficienti dell'equazione 5.18, il calcolo del gradiente delle componenti di velocità è immediato.

Capitolo 6

Risultati Numerici

In questo capitolo vengono presentati alcuni risultati numerici inerenti agli schemi presentati nel [capitolo 5](#), questi hanno permesso di validare le modifiche apportate al codice *ImmerFlow*. La validazione viene svolta attraverso comparazione con risultati sperimentali o con soluzioni analitiche a campi di moto noti. La visualizzazione dei risultati è stata svolta tramite il software Paraview [\[28\]](#), mentre i grafici sono stati creati con gnuplot [\[32\]](#).

Trattandosi di un solutore per flussi incompressibili, tutti i casi trattati saranno anche viscosi data l'assenza in natura di casi incompressibili che si possono ricondurre anche a casi inviscidi.

6.1 Cilindro Circolare Infinito Laminare

Il primo caso di studio tratta un cilindro bidimensionale circolare in regime laminare. Numerosi sono i dati sperimentali reperibili in letteratura, come anche numerose sono le simulazioni numeriche. Di particolare interesse è lo studio della scia di questo corpo, dovuta alla varia tipologia di fenomeni che questa sviluppa all'aumentare del numero di Reynolds. Tutto questo rende il cilindro circolare un soggetto adeguato alla validazione del codice.

6.1.1 Descrizione del caso

Il caso presentato in questo capitolo, si è scelto un numero di Reynolds basato sul diametro del cilindro pari a 10: per questo basso valore si osserva un campo di moto stazionario, laminare con assenza di *vortex shedding* [\[26\]](#). I risultati sperimentali, grazie ai quali si è svolto il confronto delle soluzioni ottenute fanno riferimento a [\[27\]](#).

Si consideri un caso di un cilindro immerso in aria. Trattandosi di un caso incompressibile, densità e viscosità cinematica non dipendono dalla temperatura del fluido, anzi rimangono costanti. Nella simulazione con il codice *ImmerFlow* vengono considerati i seguenti valori:

Densità [kg/m^3]	1.225
Viscosità [$Pa \cdot s$]	1.86×10^{-5}

Tabella 1: grandezze passive aria.

Il diametro del cilindro è stato scelto pari ad 1 mm , da cui è possibile calcolare il valore della velocità di infinito a monte che lo investe, mantenendo come riferimento il numero di Reynolds: 0.1518 m/s . Data la simmetria della geometria, dal punto di vista fisico, in una condizione di flusso all'infinito indisturbato costante, la soluzione rimarrà simmetrica ed invariata. Perciò si scelgono le seguenti componenti di velocità di campo libero a monte:

$$\mathbf{V} = \begin{Bmatrix} 0.1518 \\ 0 \end{Bmatrix} \text{ m/s}$$

Nel corso dell'analisi si vuole simulare un campo di moto in regime di corrente all'infinito indisturbata dalla presenza del cilindro. Si vuole anche che all'infinito a valle, a seguito del superamento del solido, la corrente abbia sufficiente spazio per riadattarsi ad una condizione indisturbata. Tenendo in considerazione tale dettaglio, posizionando il cilindro nell'origine di un sistema di riferimento cartesiano, si è scelto un dominio di dimensione $50\text{ cm} \times 50\text{ cm}$. Il cilindro è stato posizionato esattamente al centro, lasciando venticinque diametri in ciascuna direzione cartesiana del flusso.

Per quanto riguarda la griglia, si è optato per un caso con griglia uniforme in tutto il dominio per ridurre l'introduzione di errore numerico aggiuntivo dovuto alla non omogeneità delle celle. È stato richiesto al software di creare una griglia con dimensione massima pari a 0.25 mm . A partire da questo valore attraverso la filosofia *octree* in automatico ha suddiviso il dominio per bisezione fino al soddisfacimento dell'unico vincolo imposto. Non avendo imposto ulteriori condizioni alla griglia, quali massimo gradiente o affinamento con celle di minor dimensione a parete o nella scia del cilindro, la base dei quadrati risulta costante. La dimensione finale di ognuno dei singoli quadrati costituenti tutto il dominio è pari a 0.195312 mm . Il numero di cella totale è 73000 , escludendo quelle interne al corpo 65000 . Come si può vedere dalla figura 24 e dalla figura 25, la griglia è molto grossolana: il corpo è descritto quasi come un quadrato, non ci si aspetta quindi dei risultati molto accurati, ma un andamento simile per una conferma che il codice scritto sia corretto. Ad ogni modo si è utilizzata la massima finitura di griglia disponibile in termini di risorse computazionale e di risorse temporali.

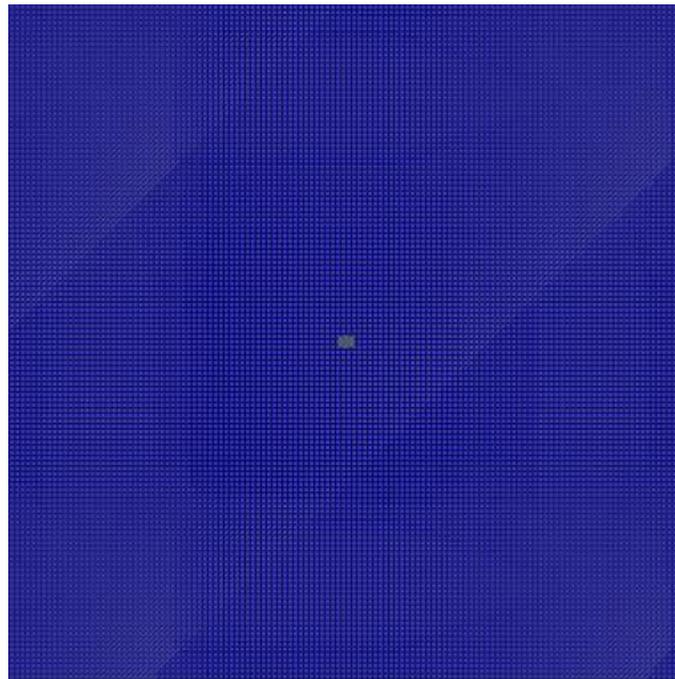


Figura 21: dominio computazionale e griglia cilindro bidimensionale.

Le condizioni al contorno imposte per il problema sono: la condizioni di velocità nulla sulla parete del cilindro, *no-slip wall*, perciò di corpo stazionario e fermo nel dominio, mentre, per quanto riguarda le condizioni al contorno sui confini del dominio computazionale, si è simulato un campo in stile galleria del vento, ovvero, condizione di *corrente libera indisturbata* a monte proveniente da sinistra del cilindro, condizione di *pressure outlet* nel bordo destro, con valore relativa rispetto alla pressione ambiente di 0 Pa , mentre per i restanti bordi si è imposto la condizione di *slip wall*.

6.1.2 Alcuni risultati

Nelle figure seguenti si riporta una serie di risultati, mostrandoti un confronto tra i vari metodi di interpolazione implementati nel corso di questa tesi, la versione originaria del software *ImmerFlow* (indicata con una linea di colore rosa) e i risultati sperimentali sopra accennati (indicati con le croci di colore nero).

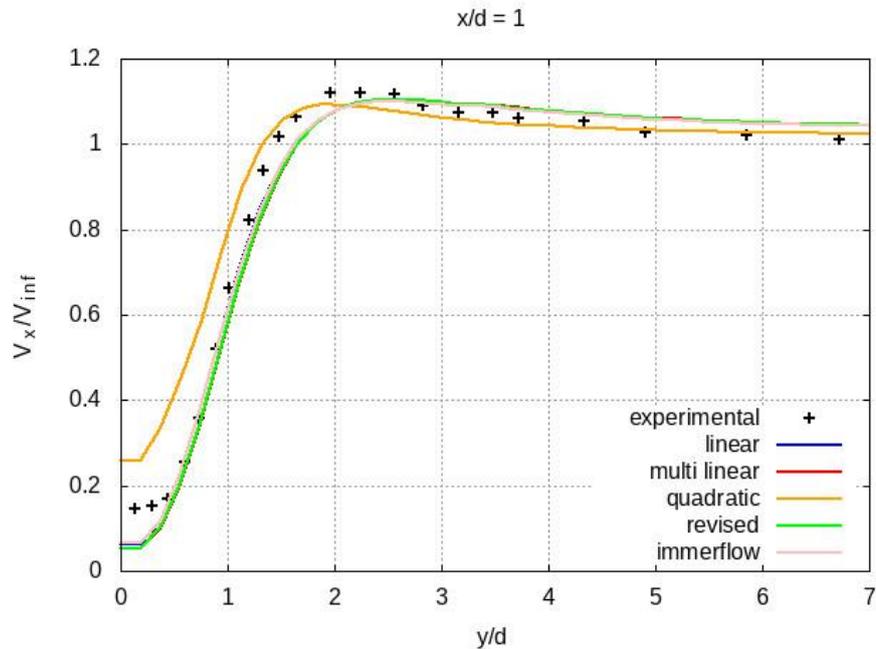


Figura 22: distribuzione della velocità lungo l'asse x , adimensionalizzata con la velocità all'infinito, ad un diametro di distanza dal cilindro, al variare della coordinata adimensionale y/d .

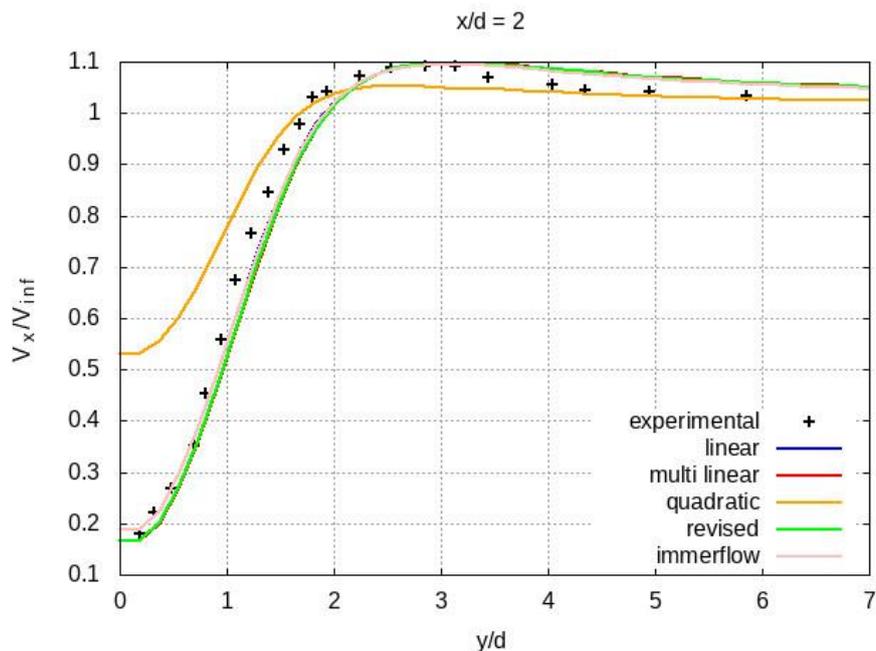


Figura 23: distribuzione della velocità lungo l'asse x , adimensionalizzata con la velocità all'infinito, a due diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d .

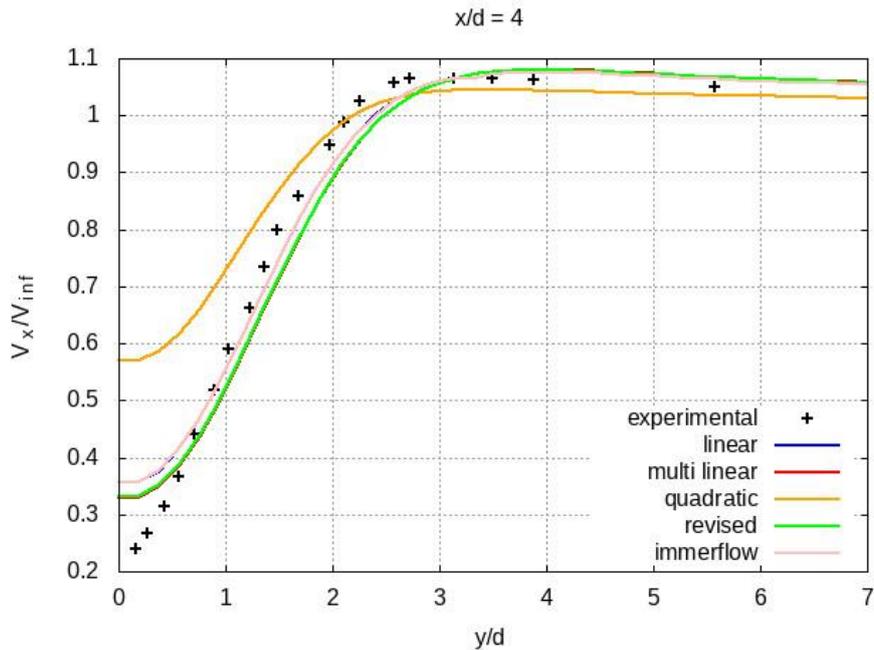


Figura 24: distribuzione della velocità lungo l'asse x , adimensionalizzato con la velocità all'infinito, a quattro diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d .

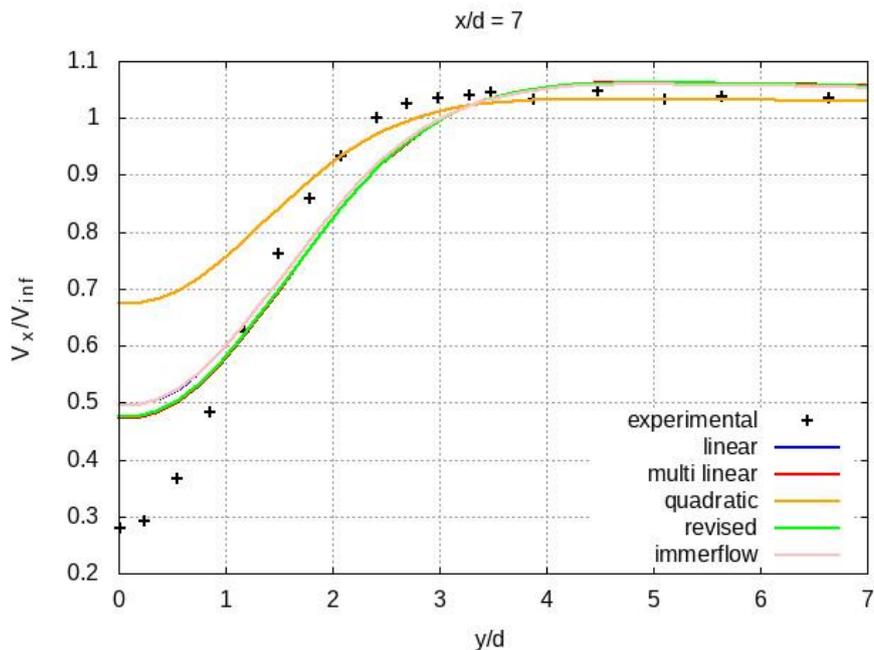


Figura 25: distribuzione della velocità lungo l'asse x , adimensionalizzato con la velocità all'infinito, a sette diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d .

Quasi tutti i risultati ottenuti presentano quasi tutti la stessa accuratezza, non discostandosi molto dalla versione originaria del codice, ad eccezione del caso con interpolazione quadratica, la quale si discosta maggiormente. Per quanto riguarda il confronto con i risultati sperimentali, si osserva buona precisione per i confronti in prossimità del cilindro, mentre allontanandosi dal corpo l'accuratezza diminuisce. Innanzitutto, questo potrebbe essere causato da una griglia rada. A seguire, un eventuale problema, potrebbe essere causato da una scarsa influenza dei termini viscosi nell'equazione della quantità di moto, approssimati in modo errato nella discretizzazione numerica.

Per eventuali simulazioni future si raccomanda di svolgere le simulazioni con una maggiore finezza di griglia, introducendo un affinamento della griglia nel seguito della geometria. Ulteriori miglioramenti possono essere introdotti anche aumentando la griglia a monte del corpo.

Lo schema di interpolazione quadratica presenta sempre i risultati peggiori tra quelli presentati, la causa che potrebbe degradare in tal modo la soluzione, si pensa, ricada nella scelta dello stencil di interpolazione che è stato scelto ed implementato.

Durante lo studio dei risultati si evidenzia che l'interpolazione lineare implementata solamente nella parte convettiva dell'equazione di *Navier-Stokes*, in quanto già presente nella parte diffusiva di *ImmerFlow*, quasi non si discosta dai risultati ottenuti con la versione originaria del codice.

Si evidenzia anche che a differenza di quanto riportato in [3] da Iaccarino, nel caso *multi lineare* non si osservano problematiche nel campo di pressione attorno al corpo, nella figura 30 si riporta un confronto tra il campo di pressione attorno al cilindro nel caso di schema *multi lineare* e *multilineare con correzione*: le immagini presentano simmetria del campo di pressione tra dorso e ventre del cilindro, confermando ancora la correttezza della soluzione.

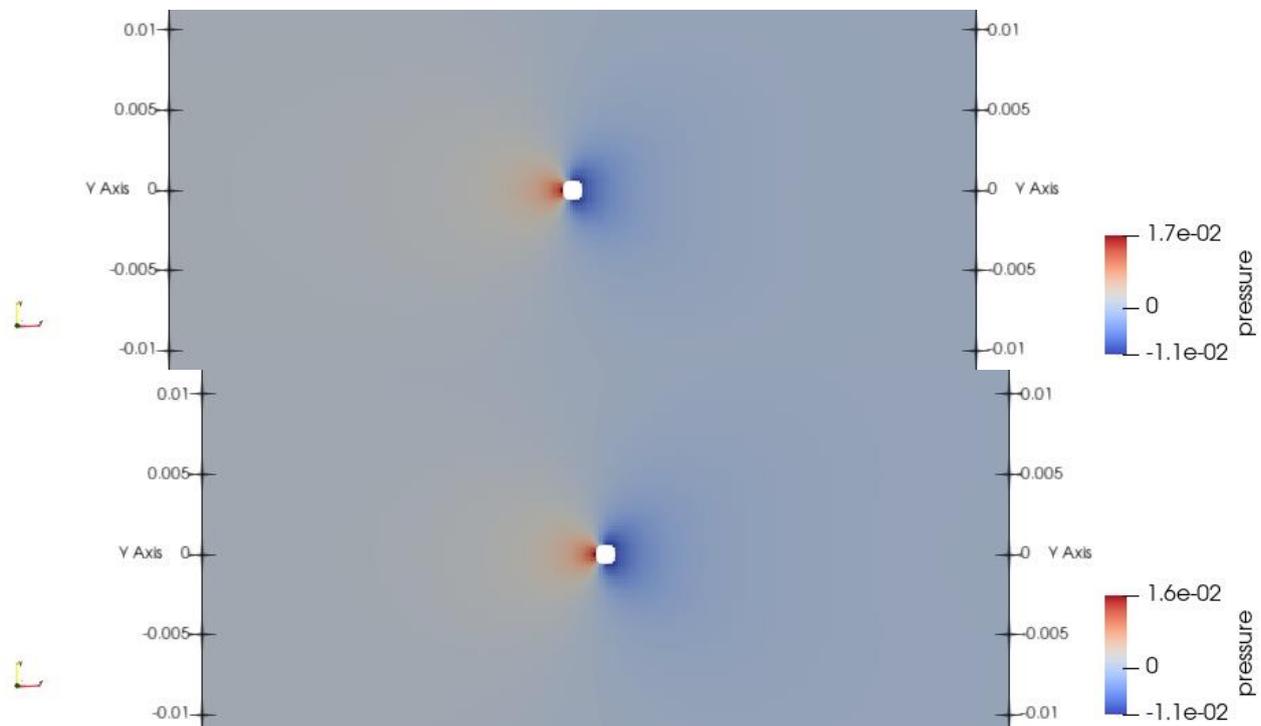


Figura 26: confronto del campo di pressione attorno al cilindro. Sopra: schema multi lineare, sotto: schema multi lineare con correzione del gradiente di pressione.

Di seguito si riportano, nella figura 31, figura 32 e nella figura 33, si riportano gli andamenti dei residui, rispettivamente, dell'equazione della quantità di moto, lungo entrambe le direzioni cartesiane e dell'equazione di continuità. Gli andamenti mostrano, il raggiungimento della condizione a regime del campo di moto.

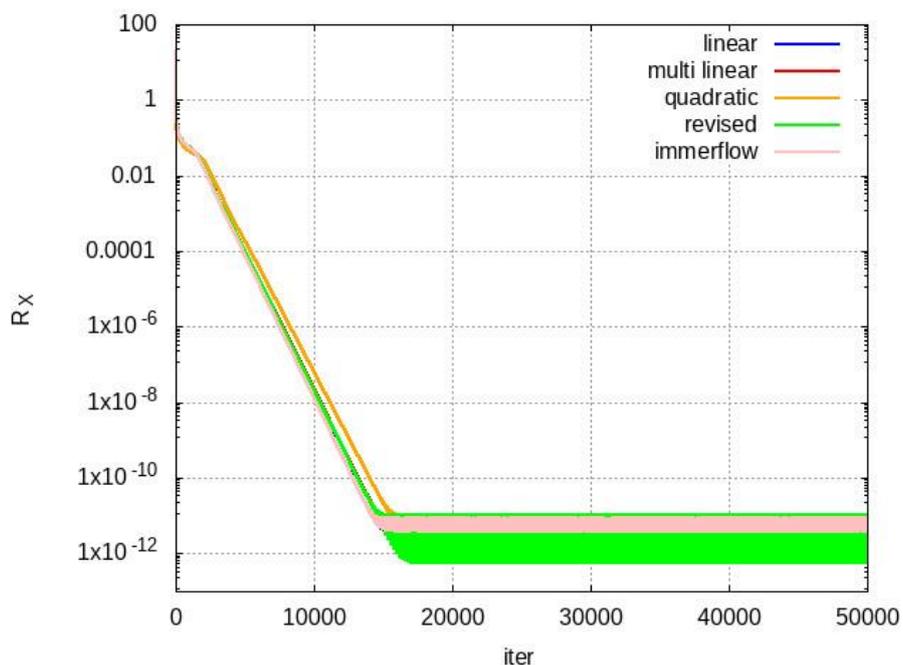


Figura 27: andamento dei residui dell'equazione di quantità di moto lungo l'asse x.

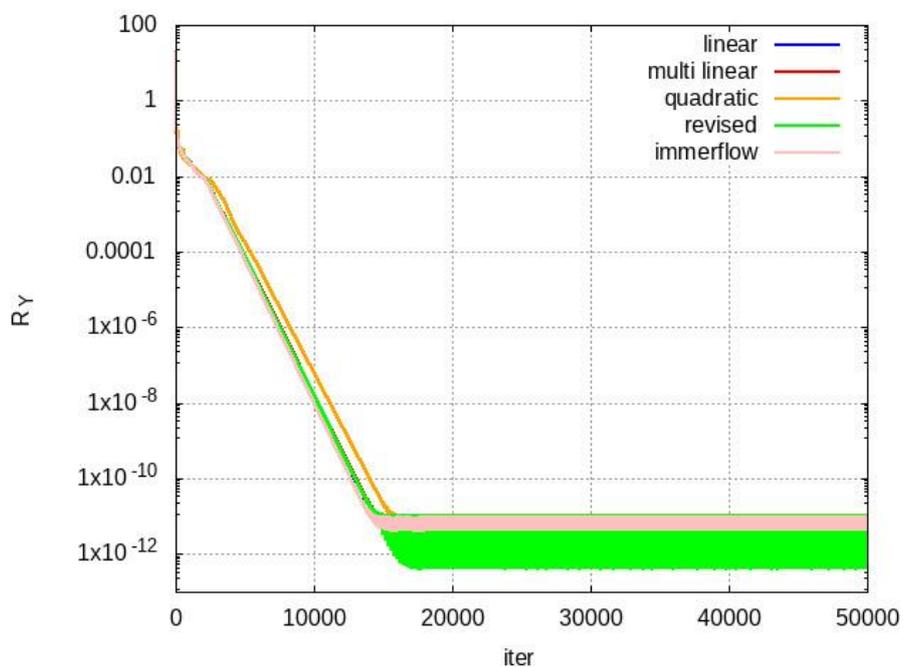


Figura 28: andamento dei residui dell'equazione di quantità di moto lungo l'asse y e andamento dei residui dell'equazione di quantità di moto lungo l'asse x.

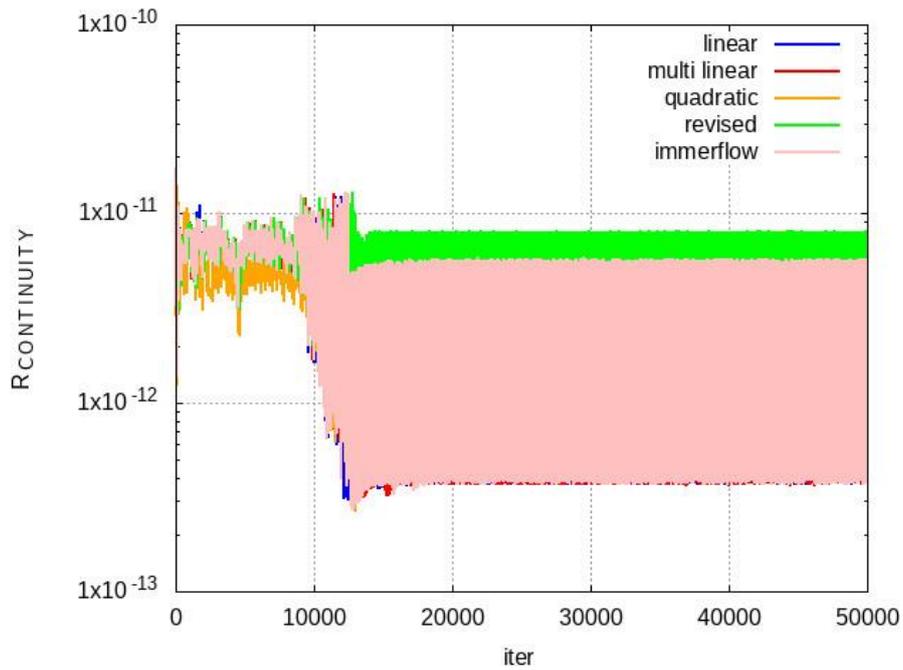


Figura 29: andamento dei residui dell'equazione di continuità.

Per quanto riguarda le immagini riportanti i residui dell'equazione di continuità si specifica che, il software forza globalmente la soluzione al rispetto della conservazione della massa ad ogni istante temporale, indipendentemente dalla soluzione ottenuta. Questo giustifica i valori bassi ottenuti nelle immagini raffiguranti tali residui. Le oscillazioni sono dovute a rumore numerico creato dalla soluzione numerica del campo.

Di seguito si riportano altre immagini di visualizzazione del campo di moto, sia di velocità che di pressione.

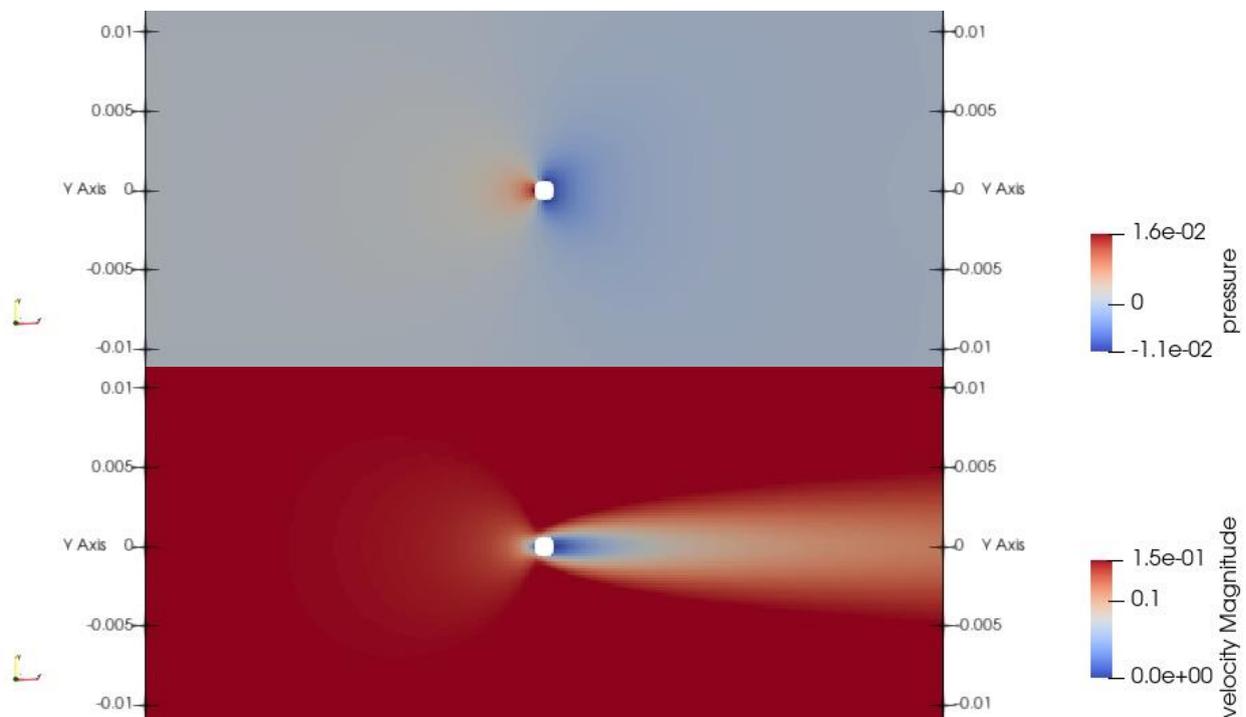
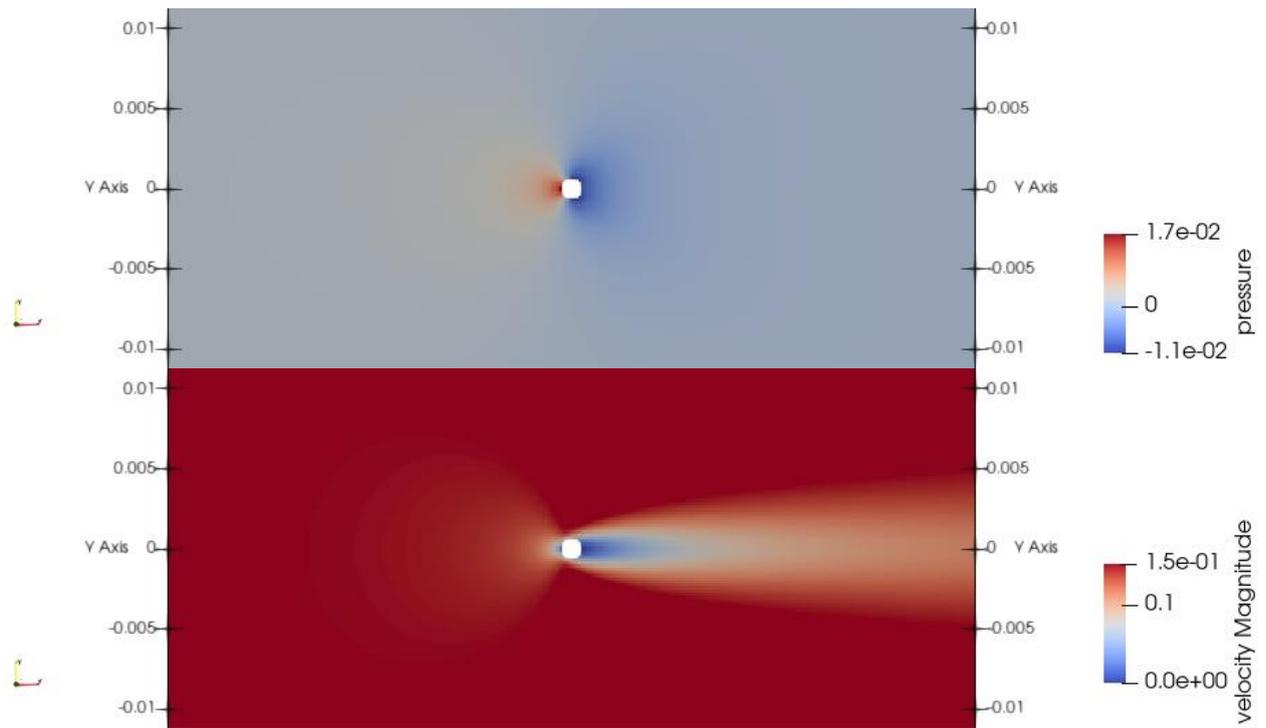
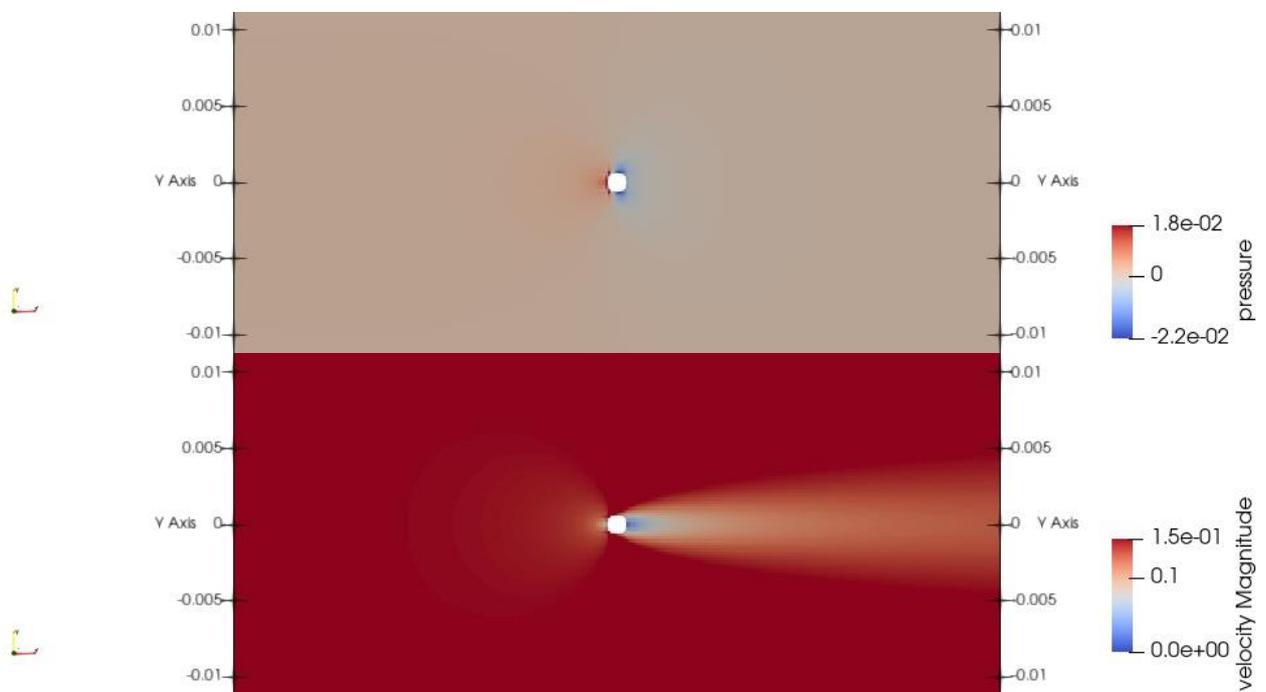


Figura 30: schema di interpolazione lineare.
Sopra: campo di pressione, sotto: campo di velocità (modulo).



*Figura 31: schema di interpolazione multi lineare.
Sopra: campo di pressione, sotto: campo di velocità (modulo).*



*Figura 32: schema di interpolazione quadratica.
Sopra: campo di pressione, sotto: campo di velocità (modulo).*

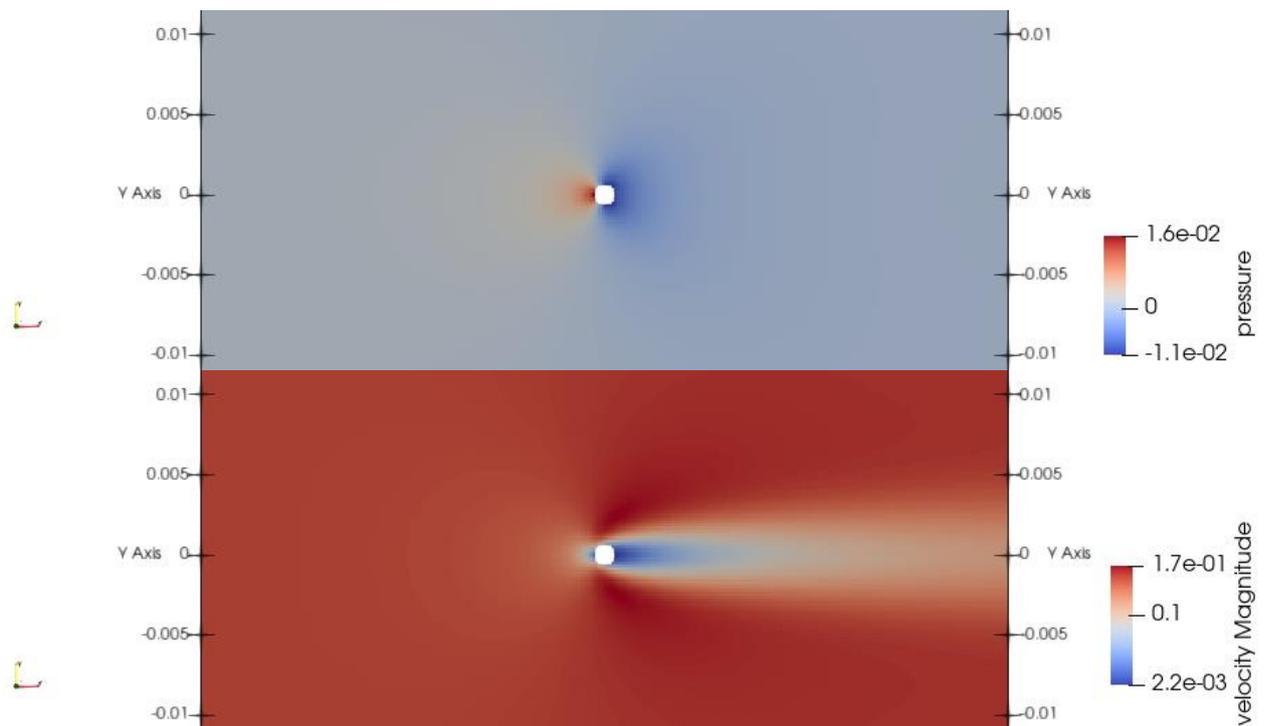


Figura 33:schema di interpolazione multi lineare con correzione tramite gradiente di pressione.

6.1.2.1 Contributo della modifica dei soli termini convettivi

Per il caso qui analizzato si è anche voluto evidenziare l'importanza dell'implementazione degli schemi di interpolazione dei termini diffusivi. Infatti, si sono svolte allora delle simulazioni nelle quali è stato modificato il solo termine convettivo, lasciando immutata la parte di codice inerente agli sforzi viscosi.

Come ci si aspettava, i risultati sono quasi del tutto coincidenti al software originario, ma si evidenzia una maggiore similitudine ai dati sperimentali dell'interpolazione multi lineare. Questo simboleggia che non sempre più si alza il grado del polinomio più si aumenta l'accuratezza dei risultati, anzi si può notare che di fronte ad un aumento del grado del polinomio, quindi del costo computazionale, la soluzione non riceve ulteriori benefici, come ci si aspettava.

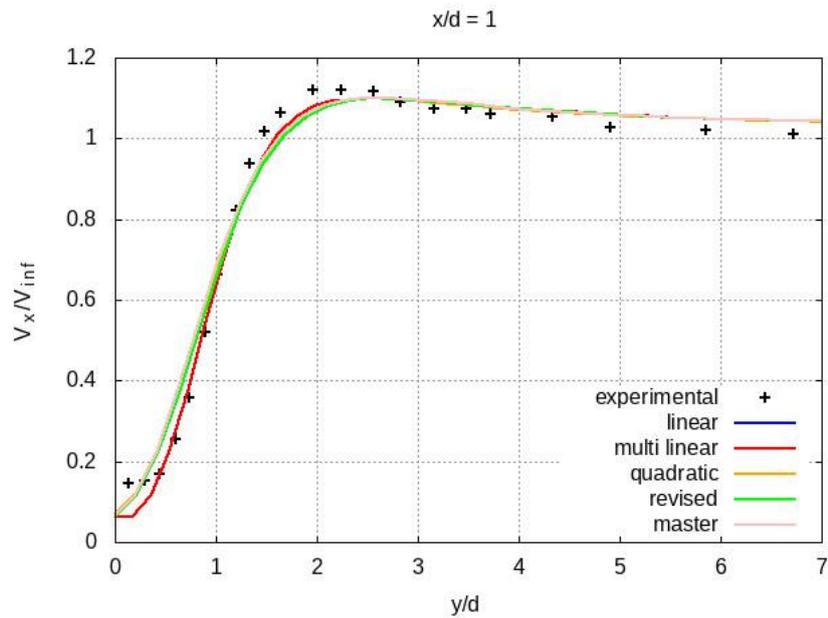


Figura 34: distribuzione della velocità lungo l'asse x , adimensionalizzato con la velocità all'infinito, ad un diametro di distanza dal cilindro, al variare della coordinata adimensionale y/d .

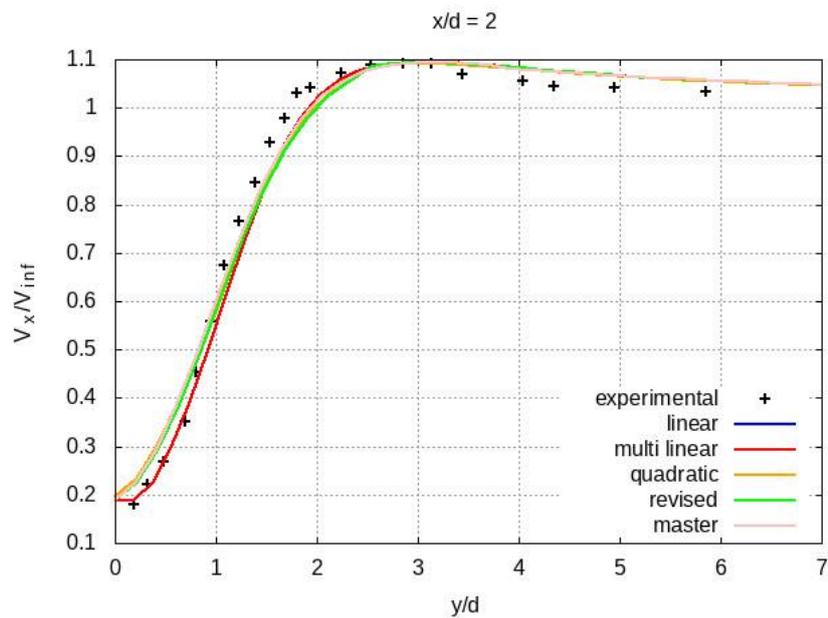


Figura 35: distribuzione della velocità lungo l'asse x , adimensionalizzato con la velocità all'infinito, a due diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d .

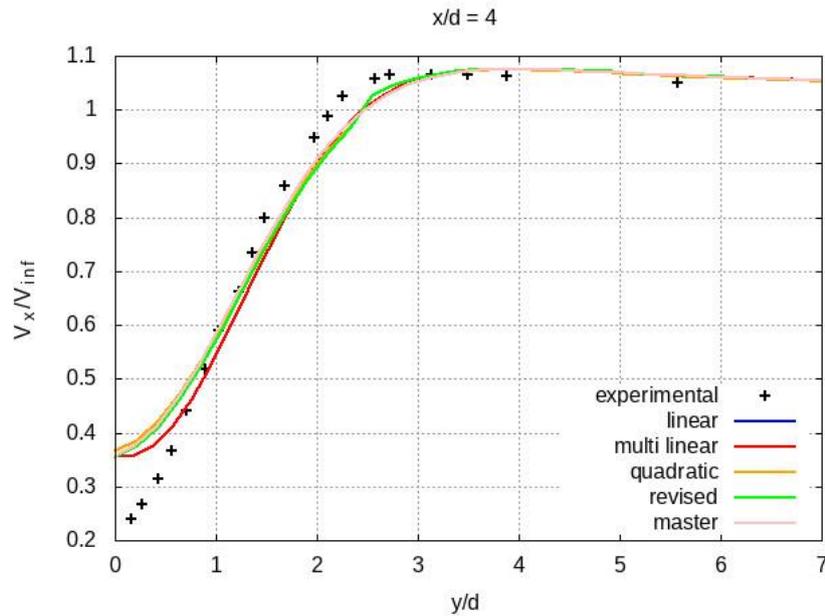


Figura 36: distribuzione della velocità lungo l'asse x , adimensionalizzato con la velocità all'infinito, a quattro diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d .

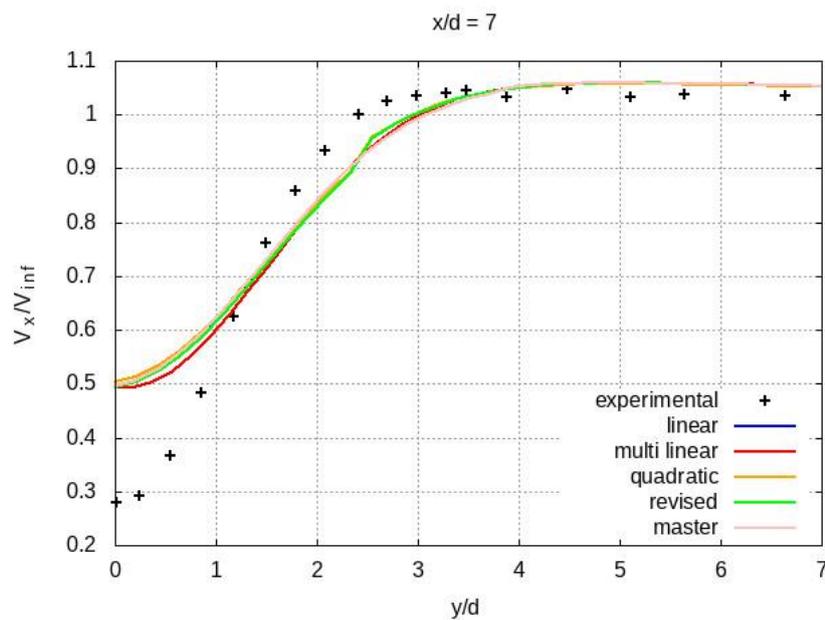


Figura 37: distribuzione della velocità lungo l'asse x , adimensionalizzato con la velocità all'infinito, a sette diametri di distanza dal cilindro, al variare della coordinata adimensionale y/d .

6.2 Lamina Piana Laminare

Il secondo caso di studio analizzato è ancora un caso bidimensionale e riguarda una lamina piana investita ad incidenza nulla, noto anche come lamina di Blasius. Di questo caso particolare è nota una soluzione analitica con la quale è possibile confrontare i risultati. In particolare, si sono svolte due analisi: la prima presentante la lamina parallela all'asse x , perciò allineata alla griglia, mentre la seconda è stata svolta con la lamina inclinata di 15 deg rispetto all'asse delle ascisse.

Per entrambe le analisi si è svolta un'analisi con numero Reynolds tale da mantenere lo strato limite in regime laminare sull'intera lamina, mantenendosi distanti dal valore di transizione alla turbolenza.

6.2.1 Lamina Orizzontale

6.2.1.1 Descrizione del caso

Si consideri, in un sistema di riferimento cartesiano bidimensionale, una lamina di lunghezza 1 m. Fissate le grandezze del fluido (si veda al [paragrafo 6.1.1](#) la [tabella 1](#), del capitolo 6), è stata scelta una velocità che permettesse di avere un campo di moto laminare.

È stata imposta una velocità di modulo 0.005 m/s parallela alla lamina, così orientata:

$$\mathbf{V} = \begin{Bmatrix} 5 \\ 0 \end{Bmatrix} \text{ mm/s}$$

Con tale modulo di velocità si è ottenuto un valore del numero di Reynolds pari a 329, rispettando il vincolo di moto fluido in regime laminare.

La dimensione del dominio computazionale è di $1.5\text{ m} \times 1.4\text{ m}$, questo è posizionato in modo da lasciare mezza corda davanti al corpo, essendo la lamina è posizionata con il primo vertice nell'origine del sistema di riferimento, il dominio computazionale vede collocato il primo vertice nella posizione $(-0.5, -0.1)\text{ m}$.

Come per il caso del cilindro, si è considerata una griglia uniforme. L'unico vincolo imposto alla sua generazione è stato, nuovamente, la dimensione massima di cella impostata ad un valore di 1 cm. Si è ottenuto un dominio discretizzato con 69000 celle, il dettaglio è mostrato nella figura 42.

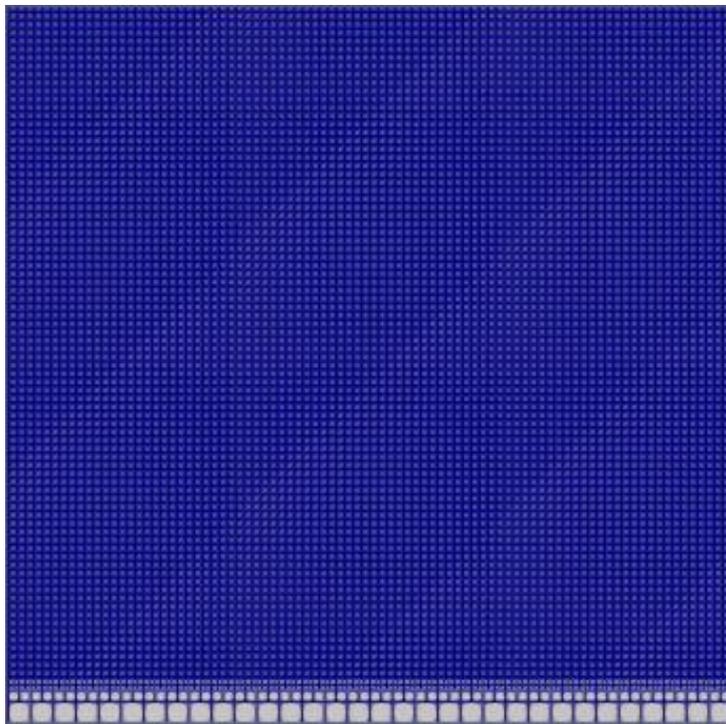


Figura 38: dominio computazionale e griglia lamina piana orizzontale.

Escludendo le celle non risolte questo numero si riduce ad un valore di 60000, si veda figura 43 le celle rimosse dal dominio fluido.

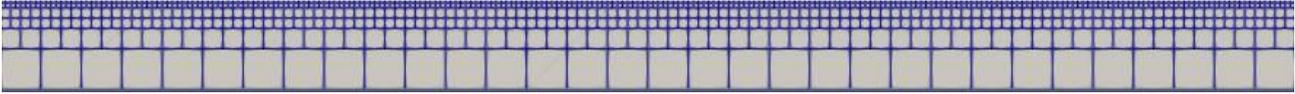


Figura 39: celle rimosse.

Nel dominio fluido la dimensione di cella è costante e pari a 5.85935 mm.

Al lato sinistro del dominio è stata imposta come condizione al contorno, il campo di *corrente libera indisturbata*, mentre al lato destro la condizione di *pressure outlet*, con valore relativo alla pressione ambiente di 0 Pa. Sulla lamina piana è stata imposta come condizione al contorno una condizione di velocità nulla, *no-slip wall*. Antecedentemente al corpo, come al bordo superiore del dominio computazionale, è stata imposta una condizione di *slip wall*. È stata effettuata questa scelta per evitare la sovrapposizione di più condizioni al contorno nel punto di intersezione, che si sarebbe creato, tra il bordo sinistro e la lamina piana, nel quale si sarebbe definito un problema mal posto a causa della presenza di più condizioni al contorno di quelle necessarie.

6.2.1.2 Alcuni Risultati

In questa sezione vengono riportati alcuni risultati. Le figure seguenti mostrano l'andamento del profilo di velocità, componente parallela alla lamina, al variare della coordinata normale ad essa. Il confronto avviene con la versione originale di *ImmerFlow* (indicata con una linea di colore rosa) e la soluzione *analitica di Blasius* (indicata con una linea di colore nero). I profili sono stati campionati nelle posizioni indicate nella tabella 6.2.

Profilo	Posizione [m]
1	0.1
2	0.2
3	0.4
4	0.7

Tabella 2: posizione dei profili nel caso di lamina orizzontale.

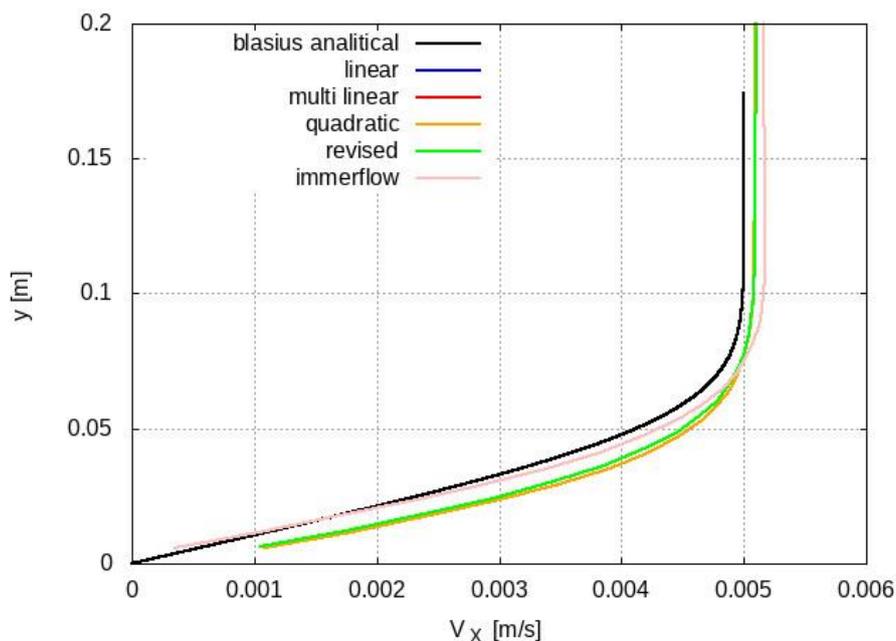


Figura 40: profilo a $x = 0.1$ m dal punto di attacco.

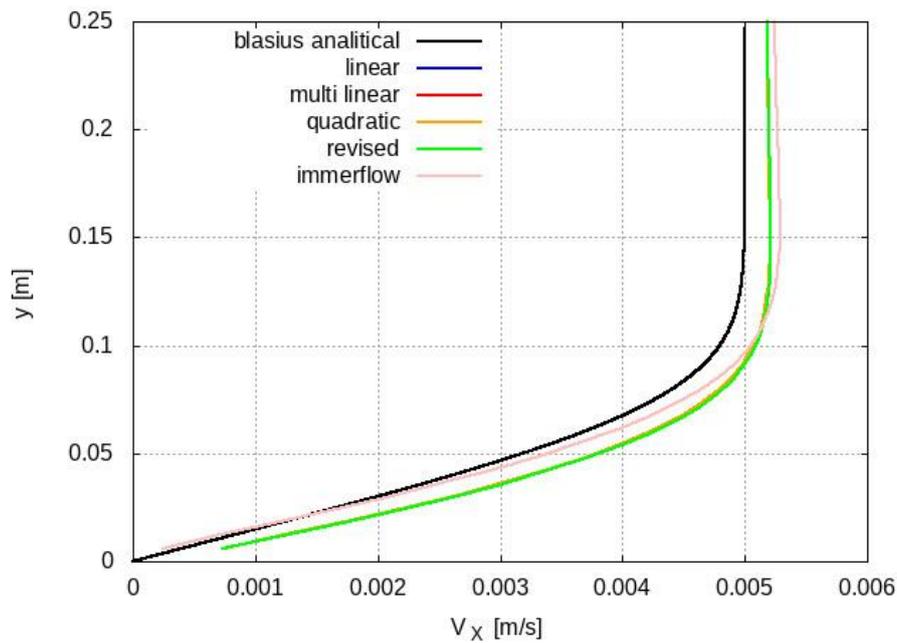


Figura 41: profilo a $x = 0.2$ m dal punto di attacco.

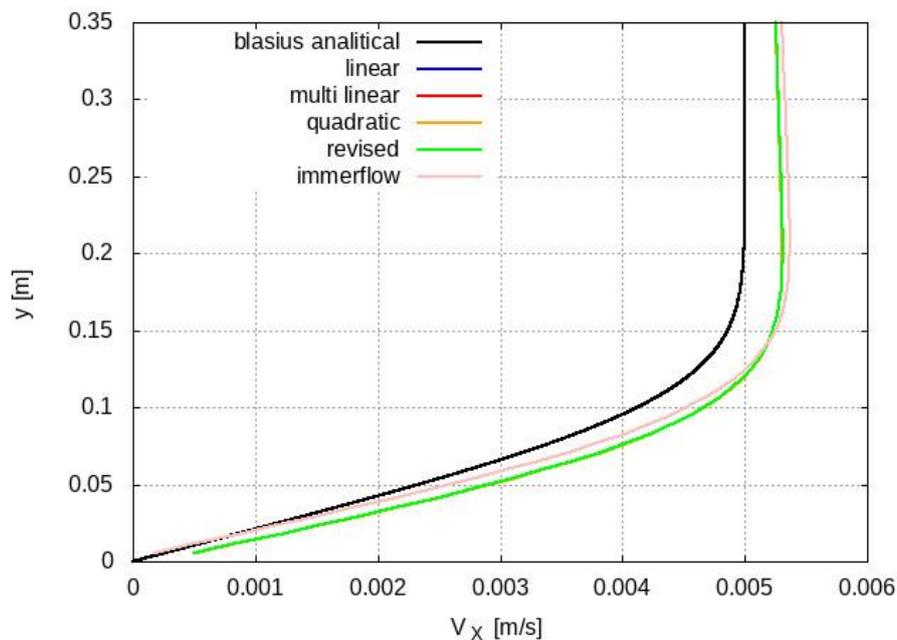


Figura 42: profilo a $x = 0.4$ m dal punto di attacco.

Fondamentale, è notare nessuno dei profili si raccorda al valore asintotico della velocità, come avviene alla soluzione analitica. Si pensa sia dovuto al fatto che l'imposizione della condizione di *slip wall* al bordo superiore del dominio facci sì che il dominio computazionale sia visto come un canale piano, in tal modo la presenza dello strato limite diminuisce la sezione di uscita del canale vista dal flusso indisturbato accelerando il flusso, rendendo il raccordo al valore asintotico di infinito impossibile.

Nonostante questo, i profili sono molto simili alla soluzione analitica, mostrando piccole differenze con essa.

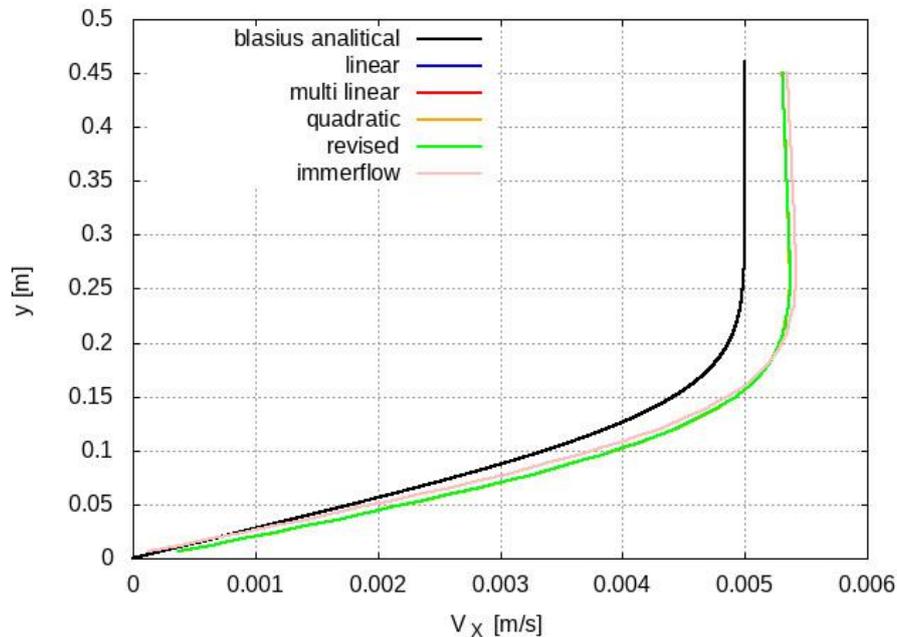


Figura 43: profilo a $x = 0.7$ m dal punto di attacco.

Oltre a non avere lo stesso andamento del profilo di velocità, anche il gradiente di velocità, in tutti i casi, non si raccorda al valore esatto del caso analitico, ma tuttavia si avvicina molto alla soluzione. Il gradiente di velocità è utilizzato per calcolare lo sforzo a parete, con il quale viene calcolata la resistenza viscosa che la lamina applica al flusso. Essendo il gradiente di velocità leggermente superiore sarà, perciò, anche parzialmente sovrastimato un eventuale calcolo della resistenza a parete.

Come per il caso del cilindro, anche qui si evidenzia che le soluzioni dello schema lineare e del software originario sono coincidenti, come lo sono anche quelle degli schemi multi lineare con e senza correzione da gradiente di pressione.

In questo caso, anche lo schema parabolico presenta buoni risultati, in quanto lo stencil di interpolazione è sempre costante, lungo il corpo, ed è uguale a quello indicato come secondo tipo nel [paragrafo 5.4.1.1](#) del [capitolo 5](#).

Di seguito si riportano, nella figura 48, figura 49 e nella figura 50, si riportano gli andamenti dei residui, rispettivamente, dell'equazione della quantità di moto, lungo entrambe le direzioni cartesiane e dell'equazione di continuità. Gli andamenti mostrano, il raggiungimento della condizione a regime del campo di moto.

Non si osserva la convergenza dei residui, ma per quanto riguarda l'equazione della quantità di moto lungo la lamina, si ha un decadimento di sette ordini di grandezza, valore più che sufficiente.

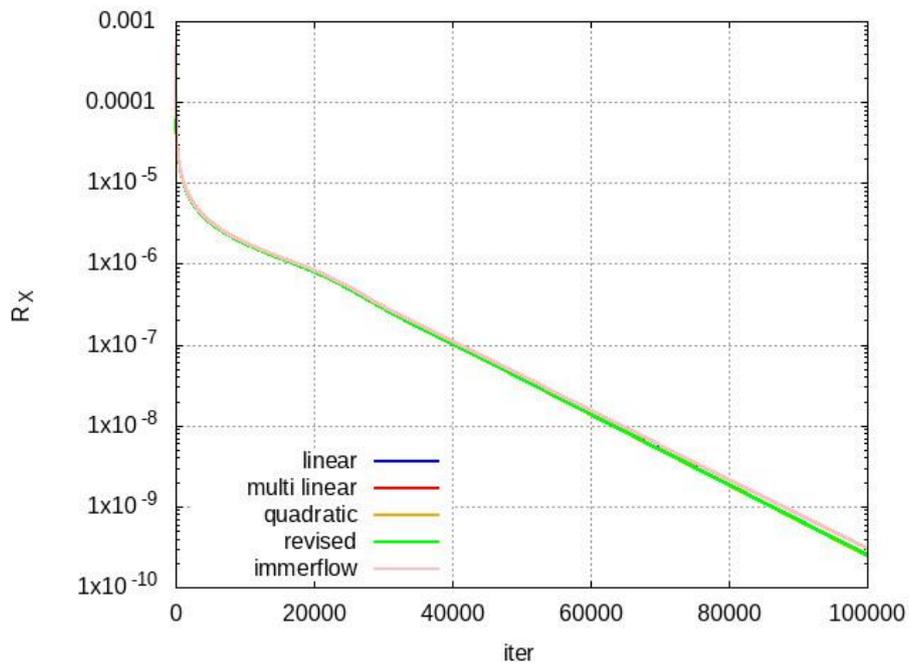


Figura 44: andamento dei residui dell'equazione di quantità di moto lungo l'asse x .

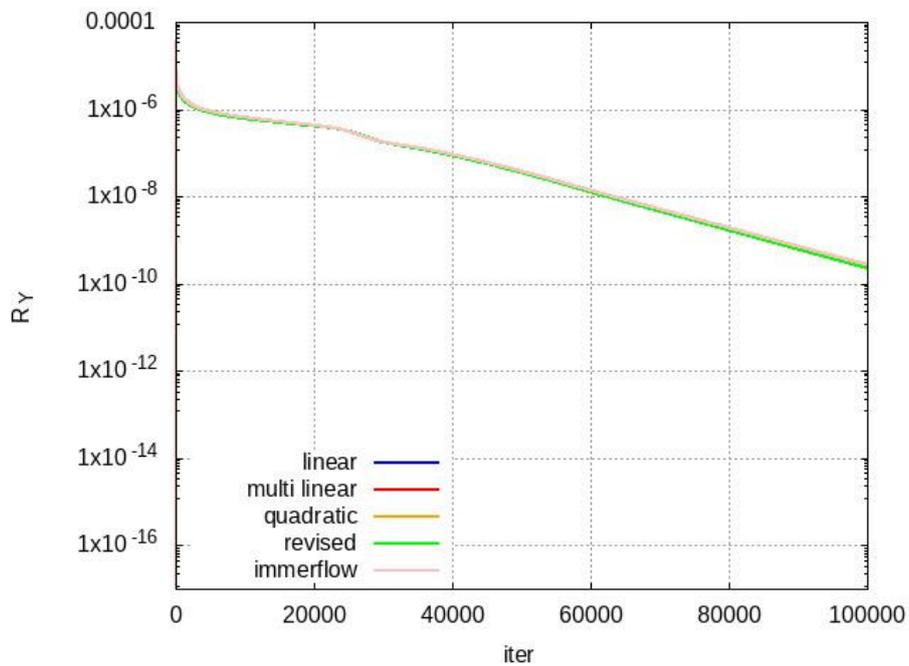


Figura 45: andamento dei residui dell'equazione di quantità di moto lungo l'asse y .

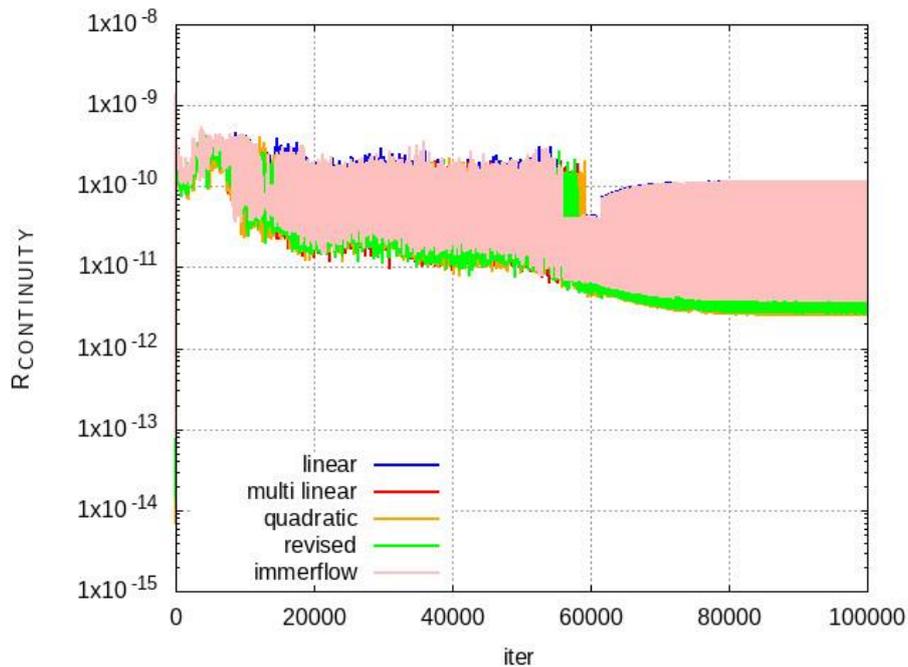
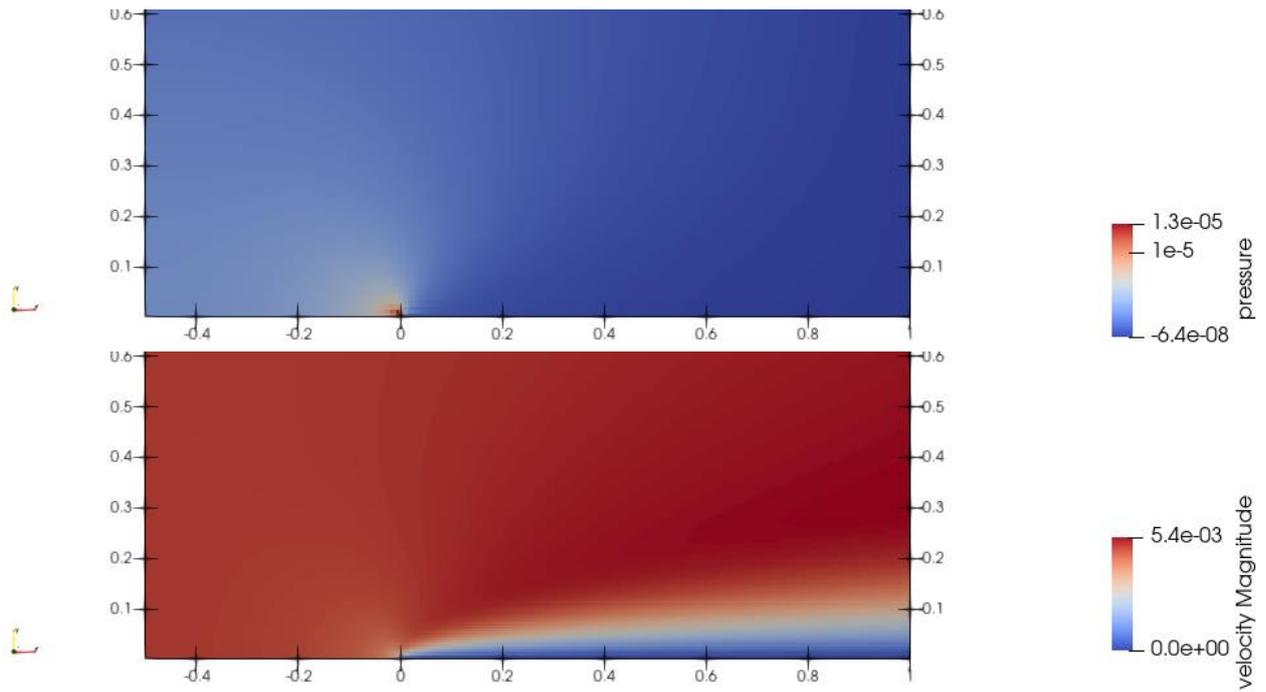


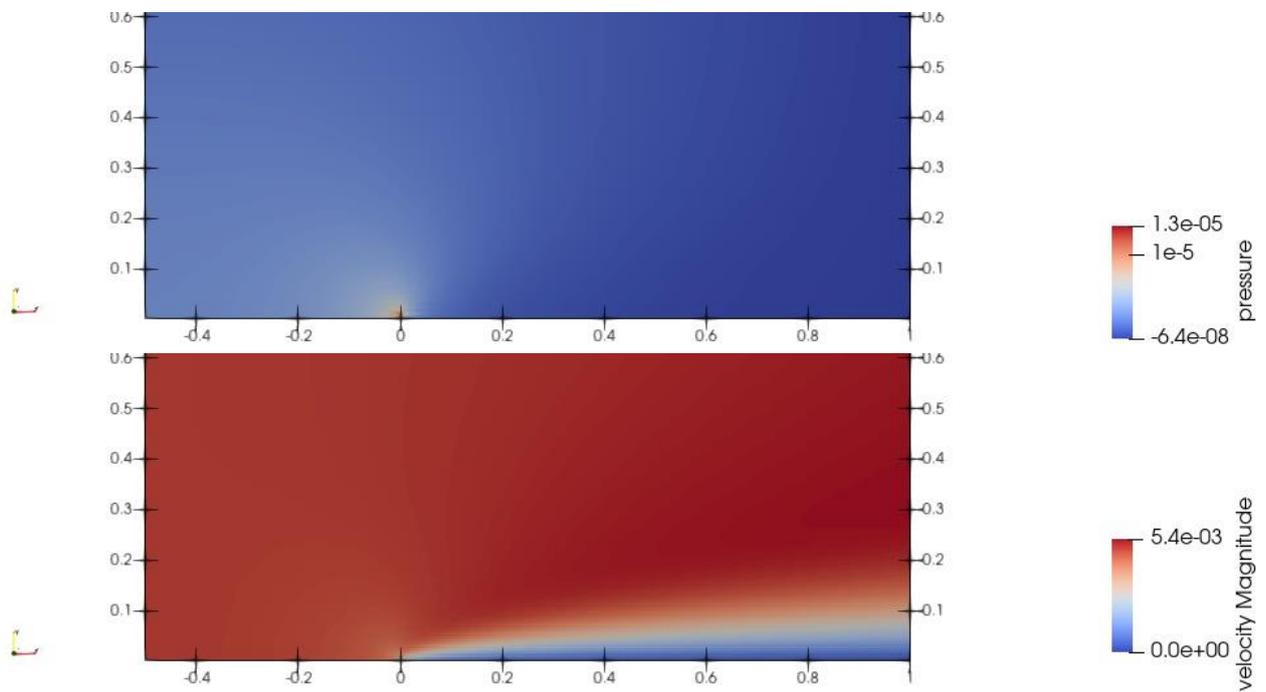
Figura 46: andamento dei residui dell'equazione di continuità.

Per quanto riguarda le immagini riportanti i residui dell'equazione di continuità si specifica che, il software forza globalmente la soluzione al rispetto della conservazione della massa ad ogni istante temporale, indipendentemente dalla soluzione ottenuta. Questo giustifica i valori bassi ottenuti nelle immagini raffiguranti tali residui. Le oscillazioni sono dovute a rumore numerico creato dalla soluzione numerica del campo.

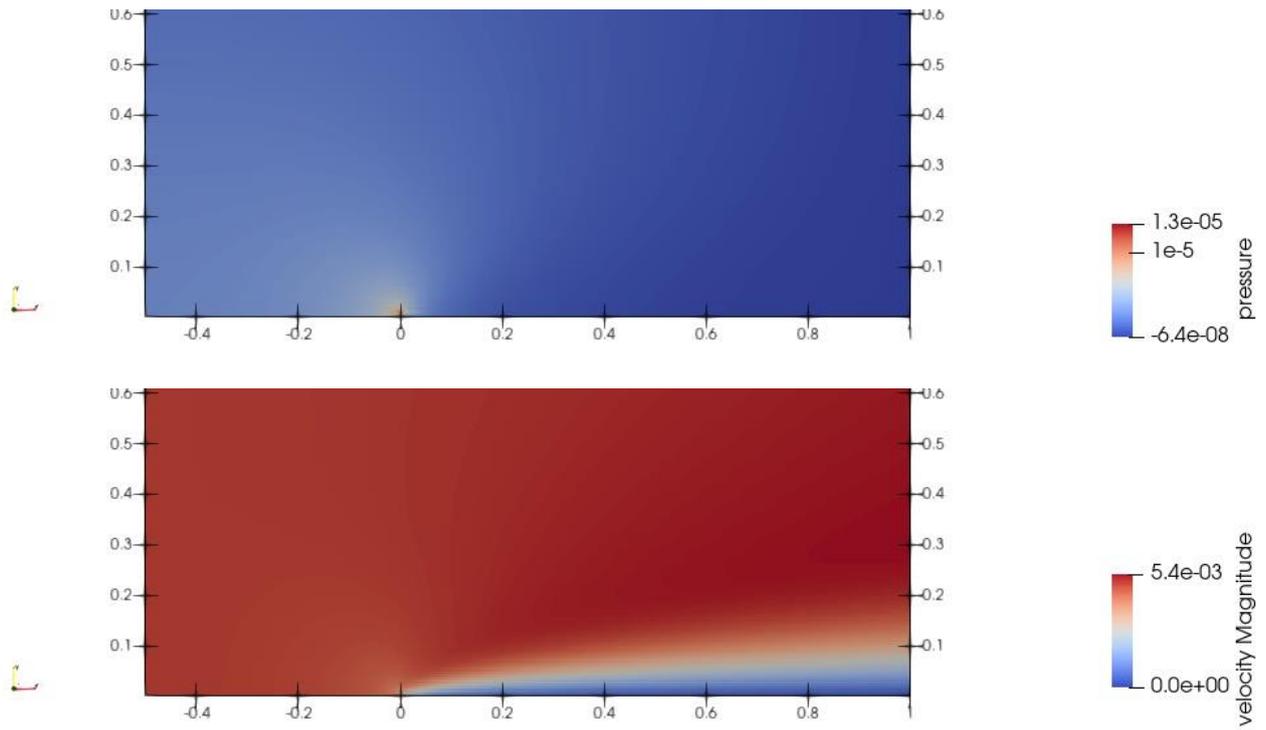
Di seguito, si riportano alcune visualizzazioni dei campi di velocità e di pressione sul dorso della lamina piana al variare dello schema di interpolazione. Per il primo, si visualizza lo strato limite laminare, con il gradiente di velocità. Per il secondo, il campo di pressione è costante sulla lamina. Questo è corretto, infatti, da approssimazioni per lo strato limite il gradiente di pressione normale al corpo è nullo, mentre data la geometria senza curvatura anche il gradiente di pressione lungo la stessa è nullo. Si visualizza solo un dipolo con variazione di pressione nelle celle in cui varia la condizione al contorno sulla parete inferiore del dominio, questo è dovuto al metodo *fractional-step* per la soluzione del sistema di equazioni implementato in *ImmerFlow*.



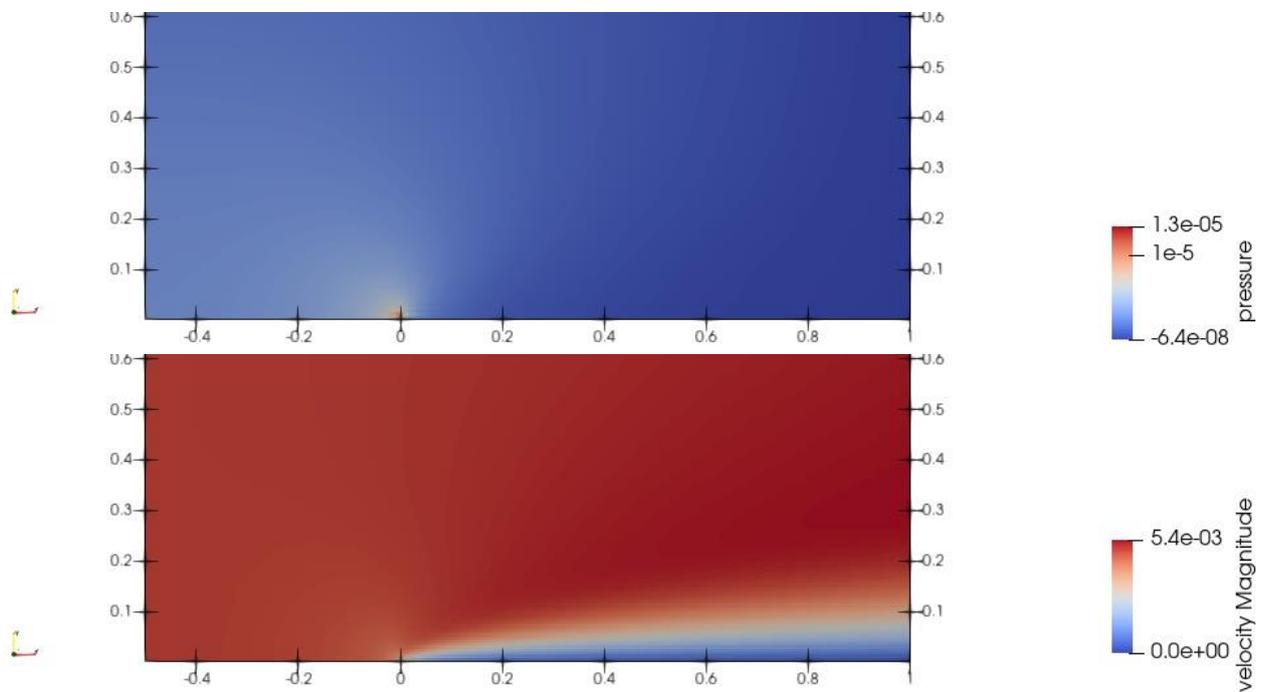
*Figura 47: schema di interpolazione lineare.
Sopra: campo di pressione, sotto: campo di velocità (modulo).*



*Figura 48: schema di interpolazione multi lineare.
Sopra: campo di pressione, sotto: campo di velocità (modulo).*



*Figura 49: schema di interpolazione quadratico.
Sopra: campo di pressione, sotto: campo di velocità (modulo)*



*Figura 50: schema di interpolazione multi lineare con correzione tramite gradiente di pressione.
Sopra: campo di pressione, sotto: campo di velocità (modulo)*

6.2.2 Lamina Inclinata

6.2.2.1 Descrizione del Caso

In analogia con il caso della lamina orizzontale, si consideri un sistema di riferimento cartesiano bidimensionale. Si consideri anche una lamina piana inclinata di 15 deg rispetto all'asse delle ascisse.

La presenza di incidenza nella lamina rispetto alla griglia forza l'utilizzo del metodo *Immersed Boundary*. A differenza del caso di lamina piana, [paragrafo 6.2.1](#), la superficie di separazione corpo fluido non è più una linea retta, ma bensì una spezzata, come nel caso di uno generico corpo immerso. In questo caso, è meglio valutabile la bontà dell'algorithm implementato.

Il numero di Reynolds scelto per questa analisi è pari a 1075, lo strato limite è ancora laminare per l'intera lunghezza della lamina.

Il campo di velocità è parallelo alla lamina con un modulo di circa $6.83 \times 10^{-3} \text{ m/s}$ e le seguenti componenti cartesiane:

$$\mathbf{V} = \begin{Bmatrix} 6.6 \\ 1.77 \end{Bmatrix} \text{ mm/s}$$

La dimensione del dominio è di 2.5 m , posizionato a partire dal vertice $(-0.19, -0.052) \text{ m}$. La lamina è posizionata nell'origine del sistema di riferimento, perciò, risulta di una lunghezza pari a 2.39 m .

La griglia considerata è uniforme nel dominio fluido. La dimensione di cella massima consentita pari a 5 mm . Si è ottenuta una griglia con 231000 volumi finiti. Il numero si riduce a 226000 se si considerano solo le celle che rientrano nel calcolo.

La figura 55 mostra in dettaglio tutto il dominio e la griglia globale. La figura 56 mostra, invece, un dettaglio sulle celle solide, perciò non risolte.

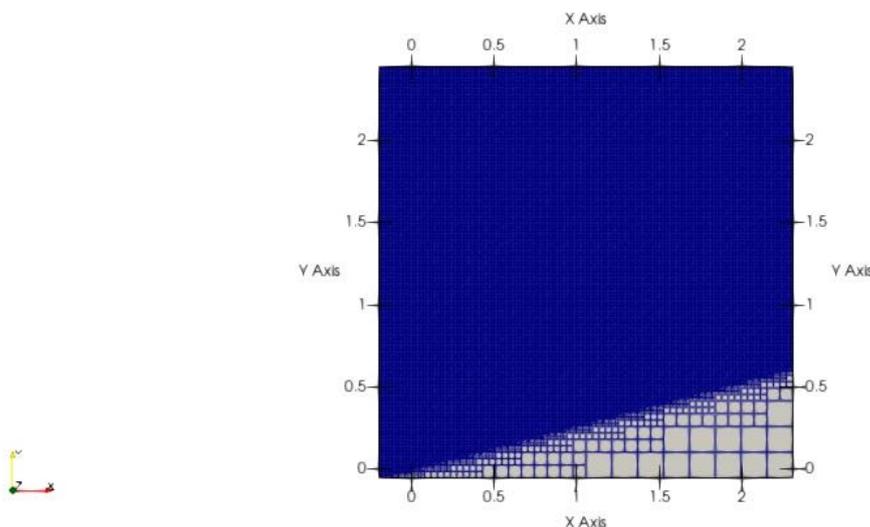


Figura 51: griglia computazionale.

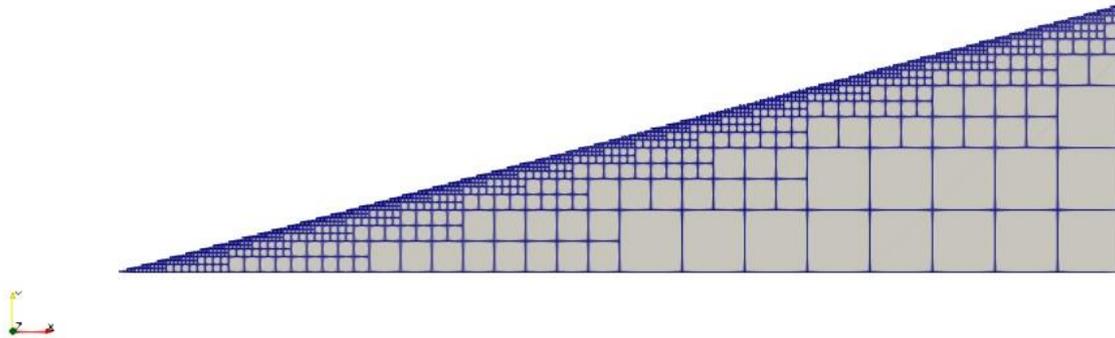


Figura 52: celle rimosse.

La dimensione di una cella fluida risulta essere di 4.88 mm.

Al lato sinistro del dominio è stata imposta come condizione al contorno, il campo di *corrente libera indisturbata*, mentre al lato destro la condizione di pressione *outlet*, con relativo alla pressione ambiente di 0 Pa. Sulla lamina piana è stata imposta come condizione al contorno una condizione di velocità nulla, *no-slip wall*. Alla parte antecedente il corpo, come al bordo superiore del dominio computazionale, è stata imposta una condizione di *slip wall*. È stata effettuata questa scelta per evitare la sovrapposizione di più condizioni al contorno nel punto di intersezione, che si sarebbe creato, tra il bordo sinistro e la lamina piana, nel quale si sarebbe definito un problema mal posto a causa della presenza di più condizioni al contorno di quelle necessarie.

6.2.2.2 Alcuni Risultati

Vengono riportati ora alcune immagini rappresentanti le soluzioni ottenute al problema appena descritto. Si svolgerà ancora il confronto con la versione originale di *ImmerFlow* (indicato con il colore rosa) e la *soluzione analitica* ottenuta da *Blasius* (indicato con il colore nero). I risultati confrontano i profili di velocità campionati a diverse coordinate lungo il dominio, riportati in tabella 6.3:

Profilo	X [m]	T [m]
1	0.1	0.1035
2	0.2	0.2071
3	0.4	0.4141
4	0.7	0.7247

Tabella 3: posizione dei profili nel caso di lamina piana inclinata di 15deg.

Come per il caso della lamina piana orizzontale, [paragrafo 6.2.1](#), anche in questo caso non si osservano dei profili precisi, ma ci si discosta dall'andamento analitico ottenuto da *Blasius*. Si presenta sempre un valore di velocità maggiore di quello asintotico.

Successivamente, i profili si inflettono cercando di raggiungere il valore asintotico della velocità imposto con la condizione di corrente libera indisturbata, imposta come condizione al contorno nel bordo superiore del dominio, ma nonostante questa piccola discrepanza si visualizza una buona concordanza tra valori i valori analitici e le soluzioni numeriche ottenute.

Anche il gradiente di velocità non è pari a quello analitico, ma comunque molto simile. Si ripresenta la stessa problematica sul calcolo della resistenza generata dalla lamina sul flusso, evidenziata nel [paragrafo 6.2.1.2](#). In particolare, si vedrebbe una leggera sovrastima del valore di resistenza, ma comunque accettabile.

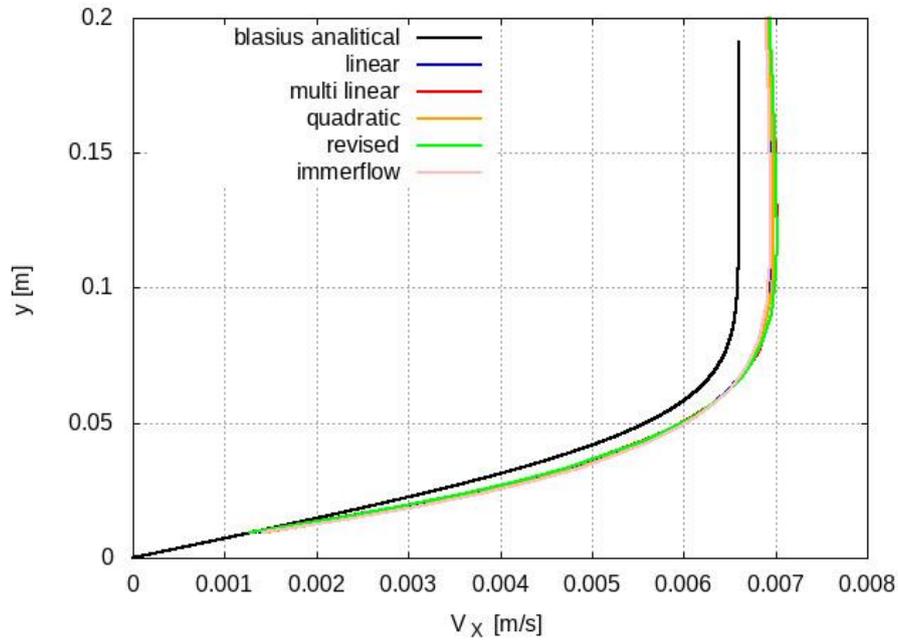


Figura 53: profilo a $x = 0.1$ m dal punto di attacco.

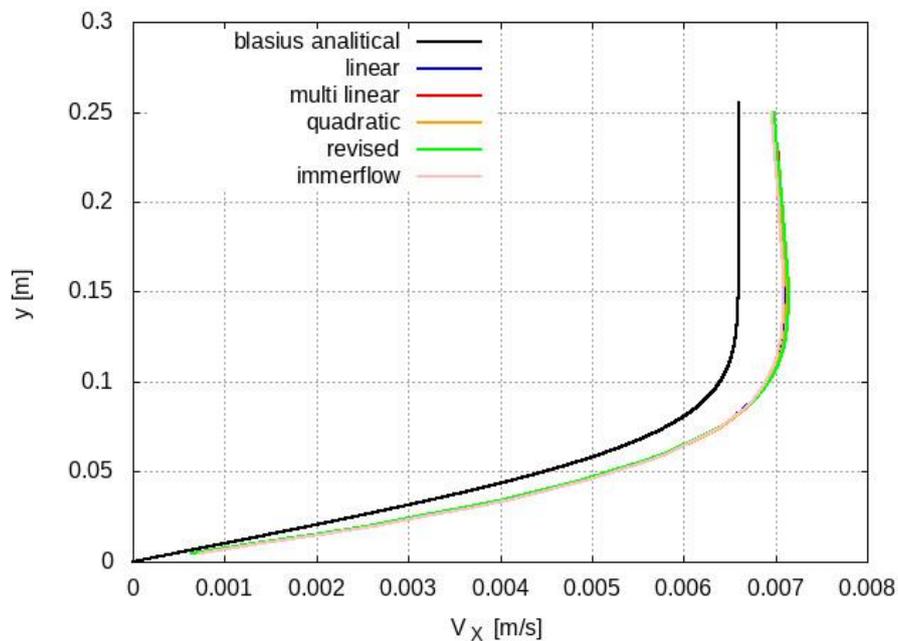


Figura 54: profilo a $x = 0.2$ m dal punto di attacco.

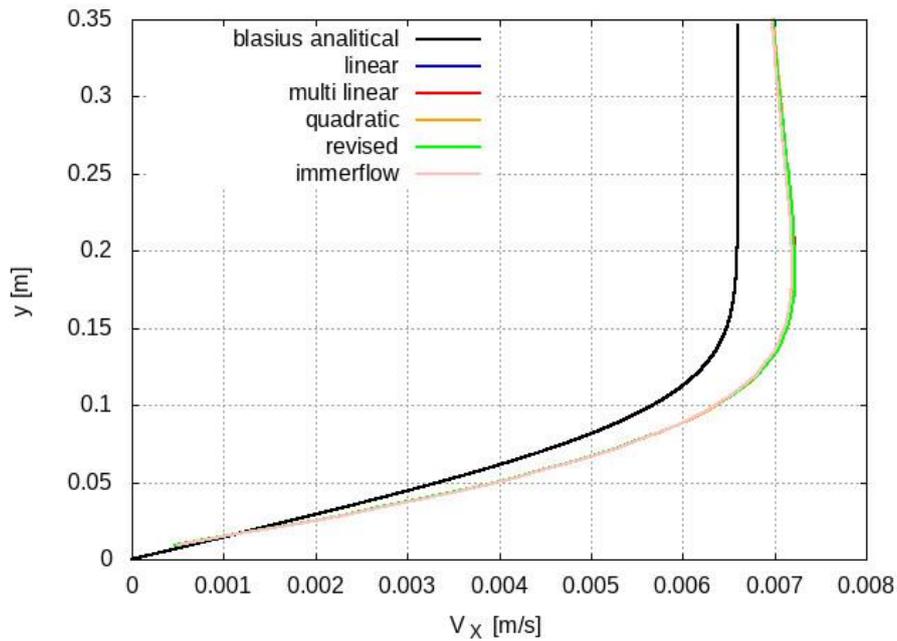


Figura 55: profilo a $x = 0.4$ m dal punto di attacco.

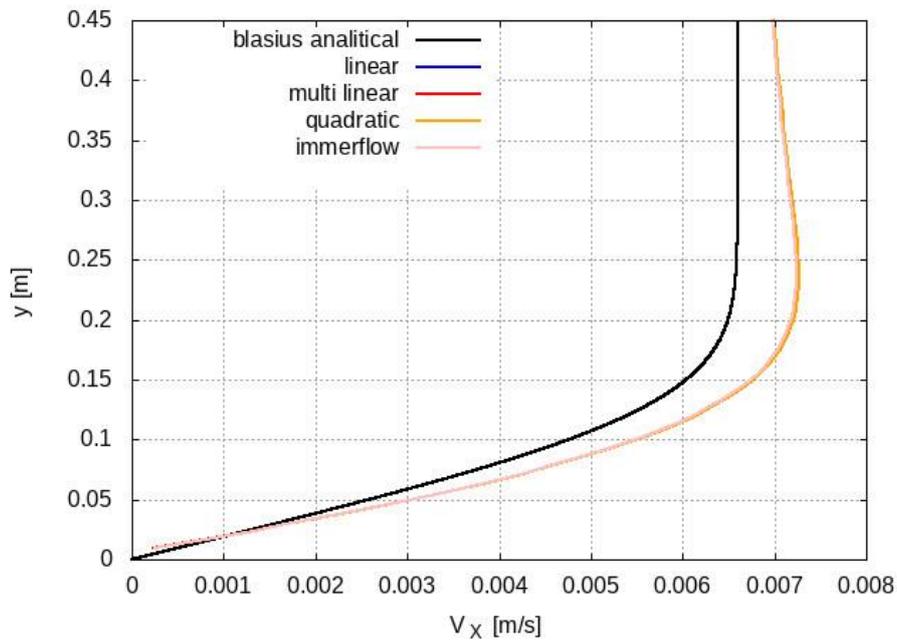


Figura 56: profilo a $x = 0.7$ m dal punto di attacco.

Qui di seguito vengono riportati gli andamenti dei residui, rispettivamente, dell'equazione della quantità di moto, lungo entrambe le direzioni cartesiane e dell'equazione di continuità. Gli andamenti mostrano, il raggiungimento della condizione a regime del campo di moto.

Non si osserva la convergenza dei residui, ma per quanto riguarda l'equazione della quantità di moto lungo la lamina, si ha un decadimento di sette ordini di grandezza, valore più che sufficiente.

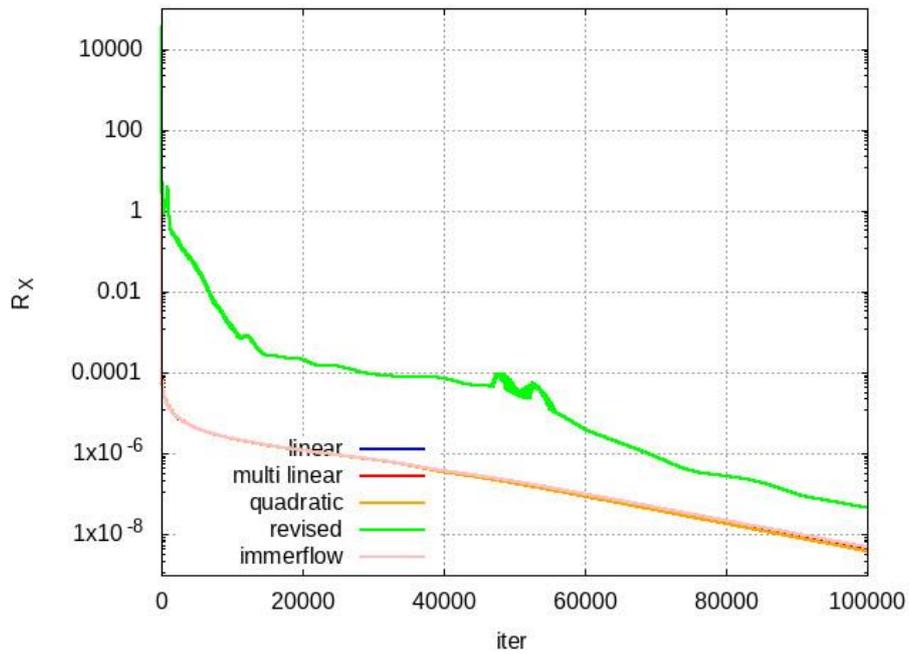


Figura 57: andamento dei residui dell'equazione di quantità di moto lungo l'asse x.

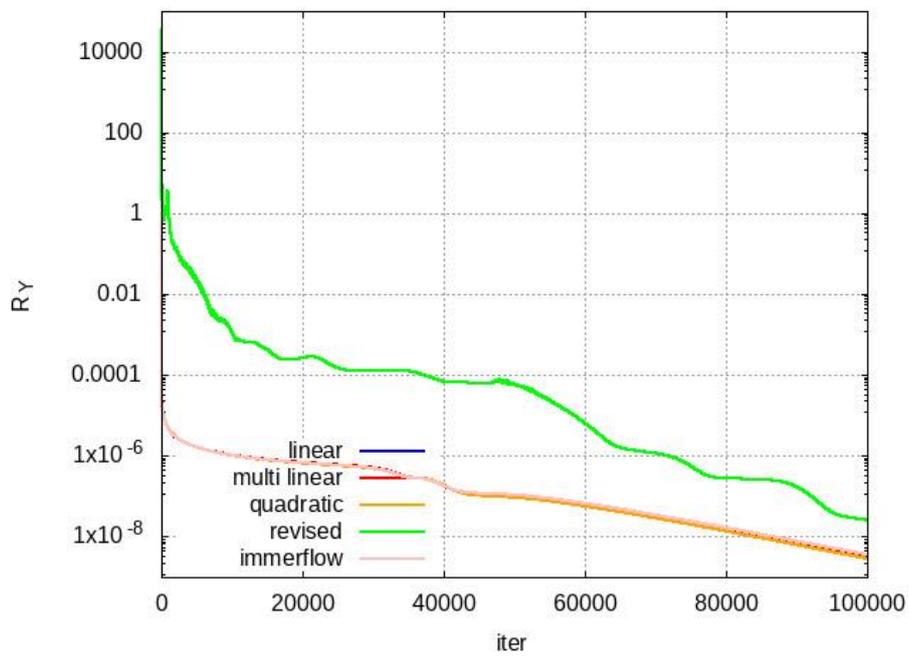


Figura 58: andamento dei residui dell'equazione di quantità di moto lungo l'asse y.

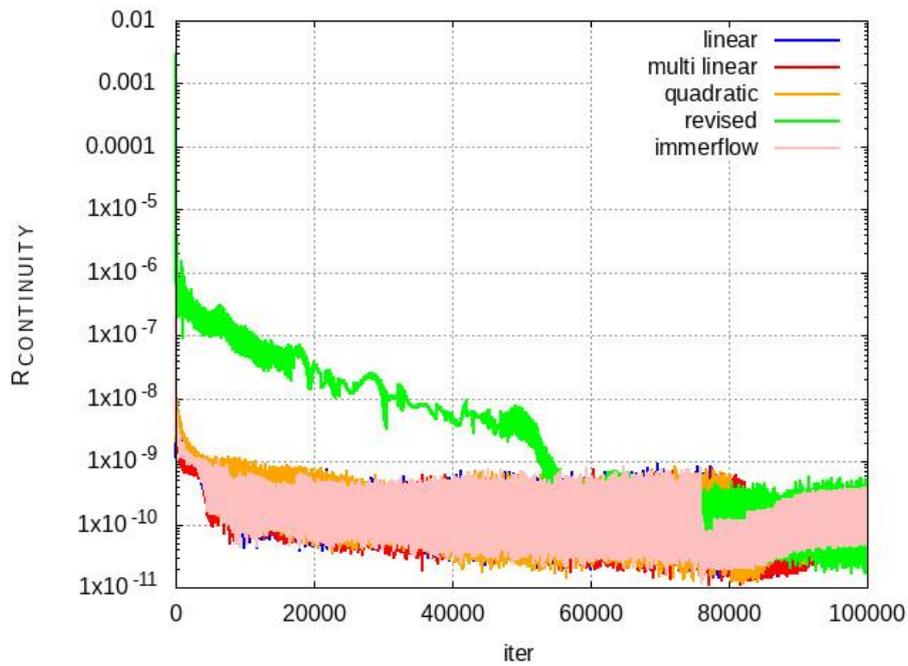


Figura 59: andamento dei residui dell'equazione di continuità.

Per quanto riguarda le immagini riportanti i residui dell'equazione di continuità si specifica che, il software forza globalmente la soluzione al rispetto della conservazione della massa ad ogni istante temporale, indipendentemente dalla soluzione ottenuta. Questo giustifica i valori bassi ottenuti nelle immagini raffiguranti tali residui. Le oscillazioni sono dovute a rumore numerico creato dalla soluzione numerica del campo.

Le prossime figure riportano i campi di velocità e di pressione sulla lamina inclinata.

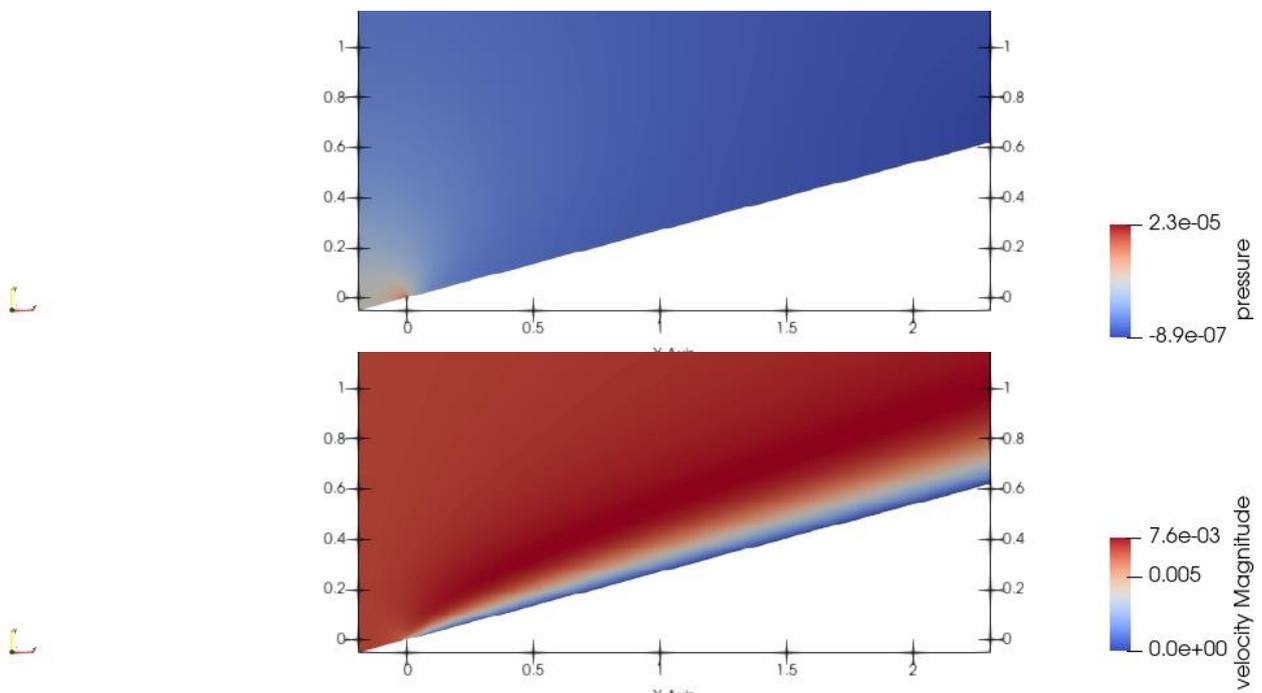


Figura 60: schema di interpolazione lineare.
Sopra: campo di pressione, sotto: campo di velocità (modulo).

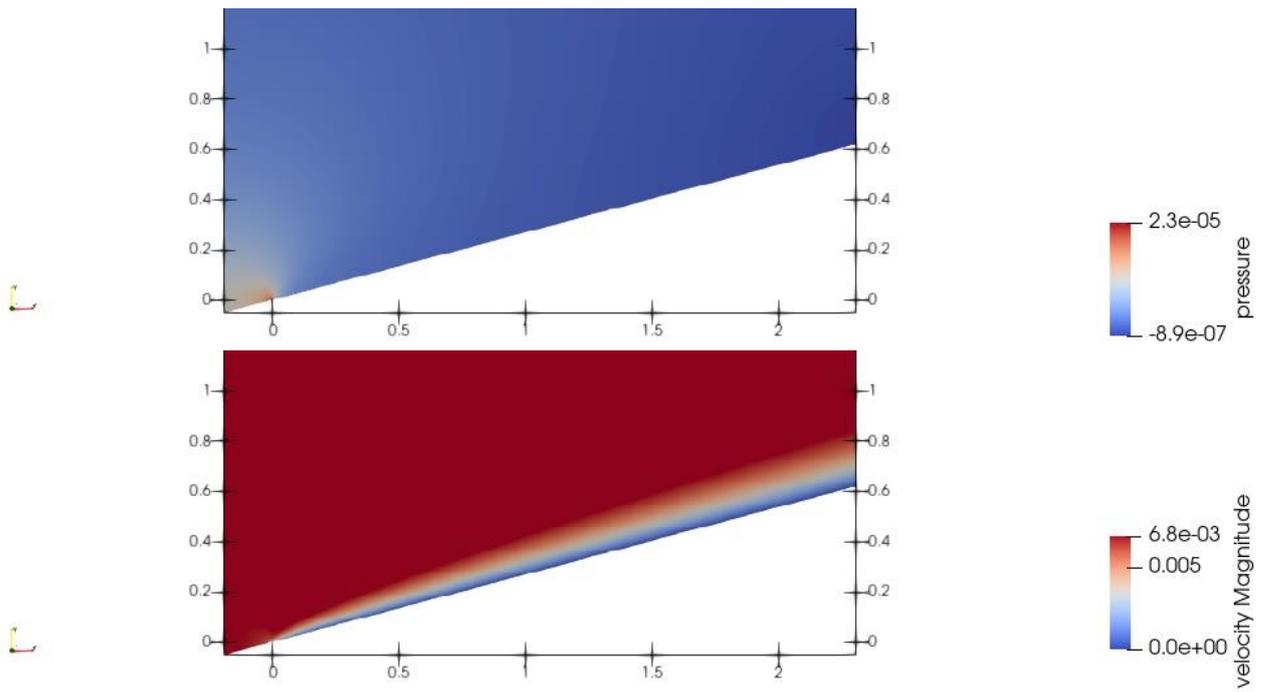


Figura 61: schema di interpolazione multi lineare.
Sopra: campo di pressione, sotto: campo di velocità (modulo).

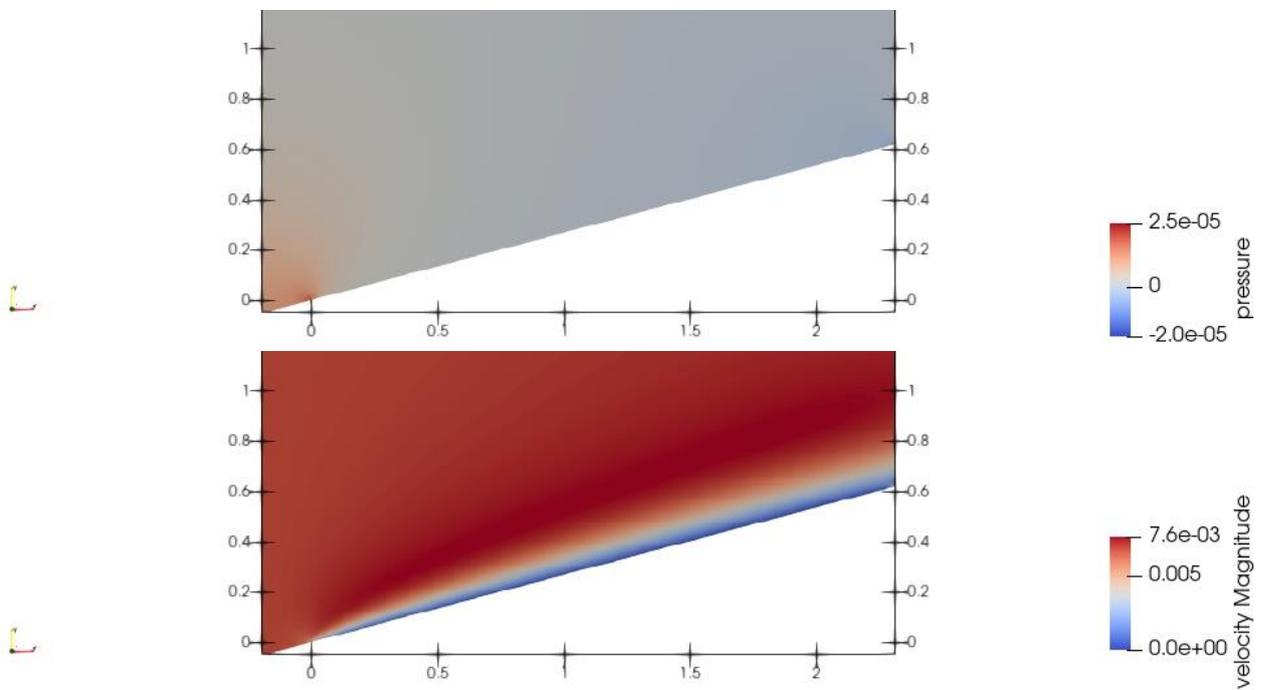


Figura 62: schema di interpolazione quadratico.
Sopra: campo di pressione, sotto: campo di velocità (modulo).

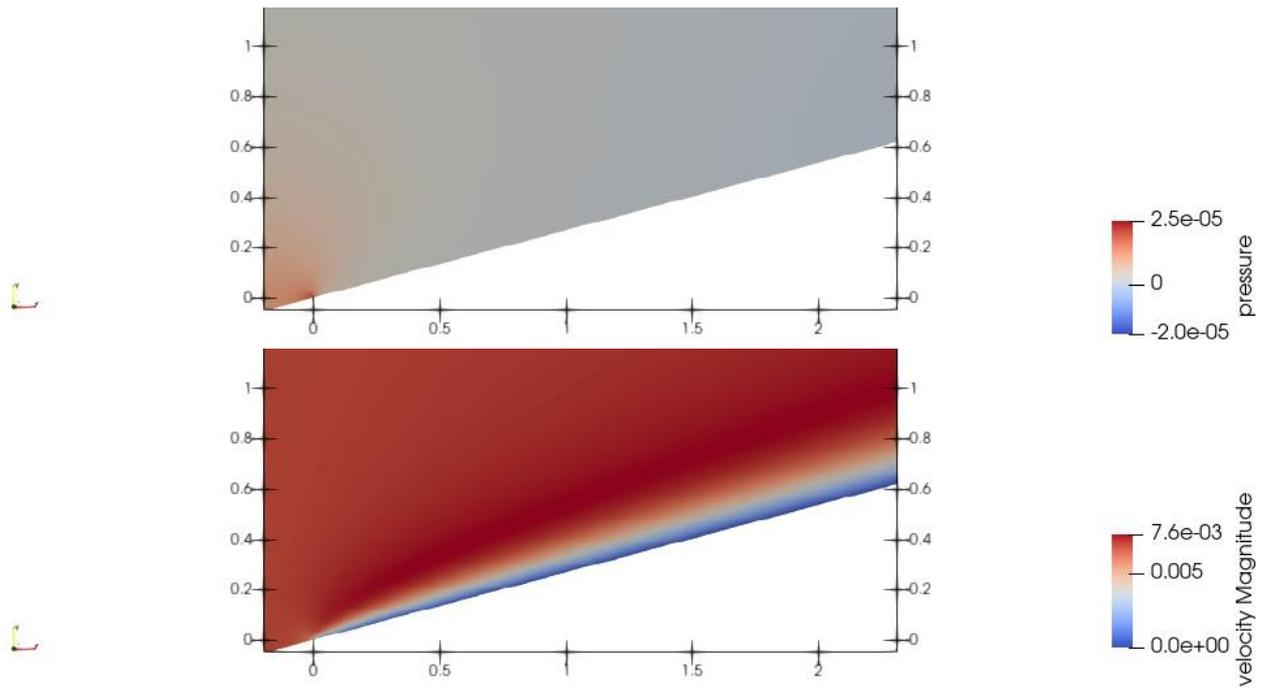


Figura 63: schema di interpolazione multi lineare con correzione tramite gradiente di pressione.
Sopra: campo di pressione, sotto: campo di velocità (modulo).

6.3 Condotto Tridimensionale Laminare

L'ultimo caso analizzato per validare il codice scritto durante questo lavoro è un caso tridimensionale. Nello specifico, il condotto è un condotto a sezione quadrata, analizzato sempre in campo laminare. Il confronto, in questo caso, è stato svolto confrontando la soluzione con la soluzione derivante dalla versione originaria di *ImmerFlow*.

6.3.1 Descrizione del caso

Essendo un caso tridimensionale in un condotto chiuso, il numero di Reynold che regola il regime di moto è basato sul diametro interno della conduttura.

Il condotto è a sezione quadrata con lato pari a 4 cm, mentre la lunghezza del condotto è di 0.4 m. Facendo riferimento a [33] dove viene trattato un caso simile in quanto a geometria, si sceglie di svolgere un'analisi con numero di Reynolds pari a 790.

Fissata dunque il numero di Reynolds, la geometria ed il fluido al suo interno viene vincolata la scelta della velocità di ingresso alla conduttura. Il modulo della velocità normale alla sezione risulta pari a 0.29988 m/s.

La geometria è stata posizionata in modo tale che l'asse del condotto fosse parallelo all'asse x del sistema di riferimento cartesiano della griglia *octree*. Si può perciò riportare il vettore della velocità in ingresso al condotto.

$$\mathbf{V} = \begin{pmatrix} 0.29988 \\ 0 \\ 0 \end{pmatrix} m/s$$

Il dominio tridimensionale è un cubo di lato anch'esso di 0.4 m, posizionato in modo tale che l'asse del condotto risulti un'asse di simmetria dello stesso, ovvero nel punto di coordinate $(-0.1, -0.2, -0.2)$ m.

Ancora una volta la griglia è uniforme con volumi cubici tutti di uguale dimensioni. Il dominio computazionale è stato suddiviso per bisezione fino al soddisfacimento dell'unico vincolo richiesto al software: che la dimensione di cella sia minore di 5 mm . Il dominio computazionale è stato suddiviso in 38000 celle, le quali scendono fino a 18000 nel dominio fluido risolto. Un dettaglio della griglia è visualizzabile nella figura 65 e figura 66. Nella prima si può osservare l'intera griglia computazionale: parte fluida risolta e parte non risolta. Dall'immagine si nota che la griglia non è uniforme come sopra riportato. Non bisogna farsi trarre in inganno perché dal punto di vista del calcolo ciò che conta è la parte fluida risolta, mentre tutta la restante griglia computazionale non è utilizzata, ma è solo un escamotage del generatore di griglia, con filosofia *octree*, per arrivare al requisito imposto. In figura 66 è mostrato un dettaglio della griglia fluida. Entrambe le immagini sono state ricavate in un piano normale all'asse del condotto, rappresentano perciò un piano generico parallelo al piano $y-z$.

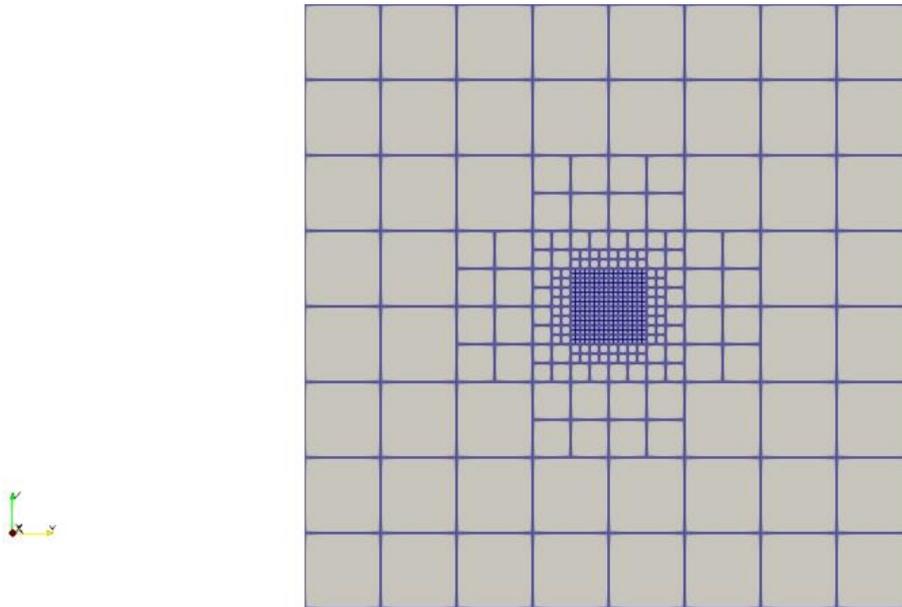


Figura 64: dettaglio della griglia totale in un piano normale all'asse del condotto.

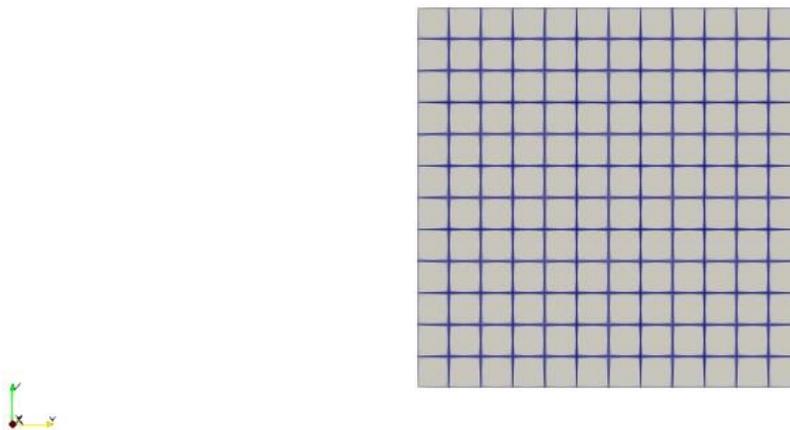


Figura 65: dettaglio della griglia nella regione fluida del condotto. La sezione è presa normale all'asse del canale.

Le condizioni al contorno imposte al problema sono, come per i casi precedenti: *corrente libera indisturbata*, *pressure outlet* e *no-slip wall*. La prima è applicata all'ingresso del canale individuato sul lato ovest del dominio. Invece, la condizione di *pressure outlet* è imposta al lato est, con un valore pari a 0 Pa . La condizione di *no-slip wall* è applicata alle pareti del dominio. Ai rimanenti due bordi del cubo è imposta la condizione di *slip wall*, questa è solo una condizione fittizia in quanto il problema è risolto internamente al canale, il flusso non vedrà mai tali condizioni al contorno.

In figura 67 è mostrata la geometria del canale analizzato in questo paragrafo.



Figura 66: geometria del tubo analizzato.

6.3.2 Alcuni Risultati

Nelle immagini seguenti si riportano alcuni risultati ottenuti dalle simulazioni del caso appena discusso svolte con i vari metodi di interpolazione e confrontati con il solutore *ImmerFlow* nella sua versione originaria (indicato con una linea rosa).

I profili mostrati riportano l'andamento della velocità assiale in due sezioni trasversali in diverse posizioni:

profilo	x [m]
1	0.20
2	0.25

Tabella 4: posizione dei profili campionati nel condotto tridimensionale.

Le immagini riportano come ascissa la velocità assiale adimensionalizzata con la velocità in ingresso, mentre come ordinata la coordinata z adimensionalizzata nel seguente modo $z^* = (z - z_{\min}) / (z_{\max} - z_{\min}) - 1$.

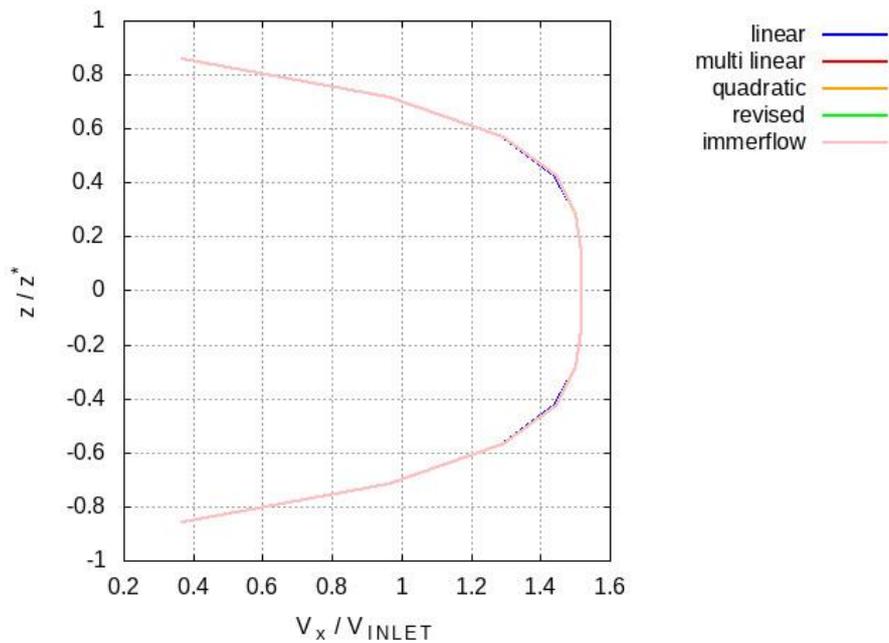


Figura 67: profilo in $x = 0.2$ m.

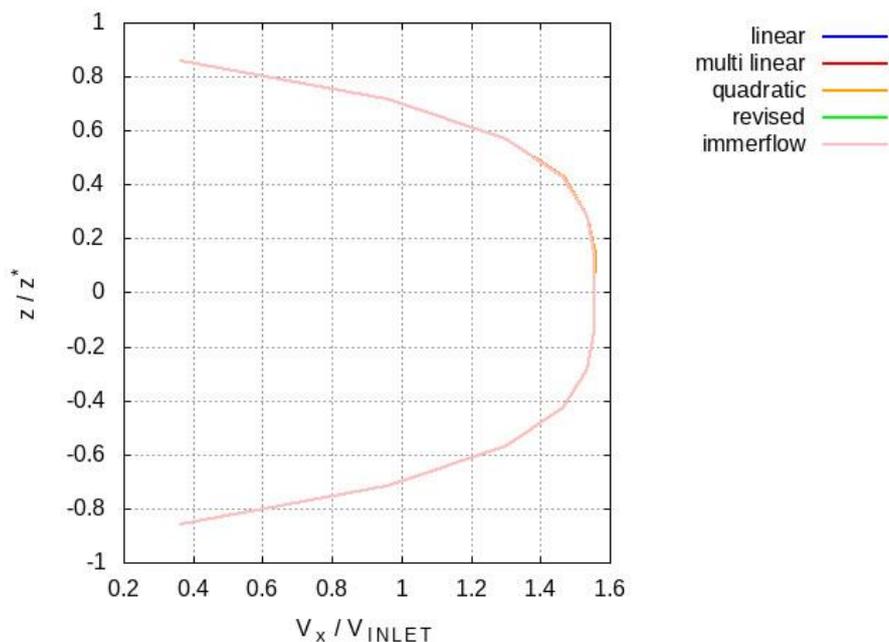


Figura 68: profilo in $x = 0.25$ m.

Tutti i profili sono simili simboleggiando la bontà di quanto implementato in questa tesi e della precedente versione. È evidente come tutti gli schemi riescano a catturare al meglio la presenza dello strato limite vicino alle pareti solide del condotto tridimensionale. Non si raggiunge il valore di velocità nulla, ma questo non è un errore, infatti, non è mai presente una cella sulla superficie del corpo. Al centro del condotto, la velocità risulta essere maggiore del valore di ingresso. Questo è corretto, infatti, procedendo lungo l'asse del canale si osserva il continuo inspessirsi dello strato limite, portando la sezione vista del fluido "indisturbato" a ridursi e causandone un'accelerazione.

Per ultimi vengono mostrati i residui delle equazioni di quantità di moto lungo l'asse x , lungo l'asse y , lungo l'asse z e i residui dell'equazione di continuità. Tutti questi simboleggiano il raggiungimento della convergenza della soluzione.

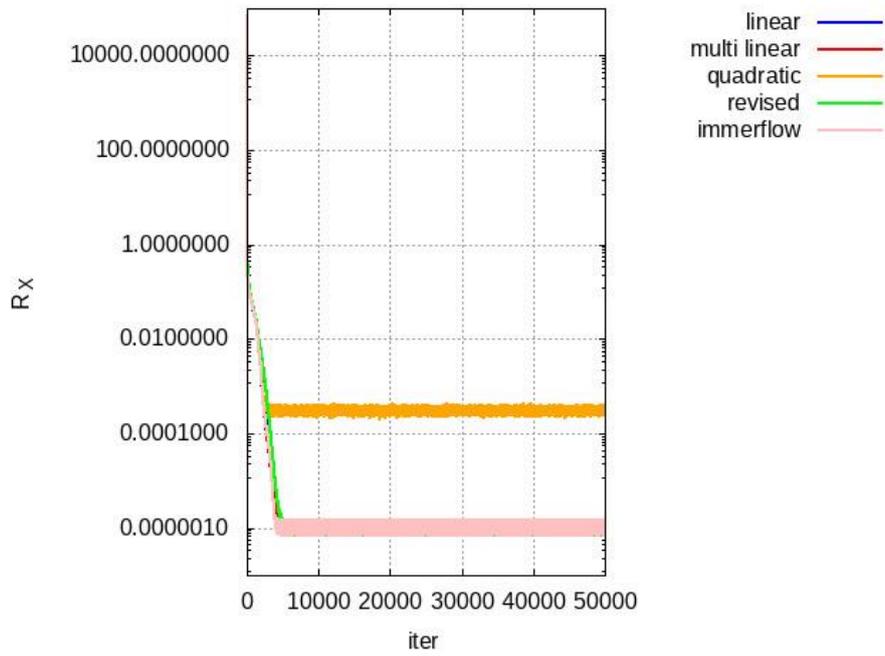


Figura 69: residui dell'equazione di quantità di moto lungo l'asse x.

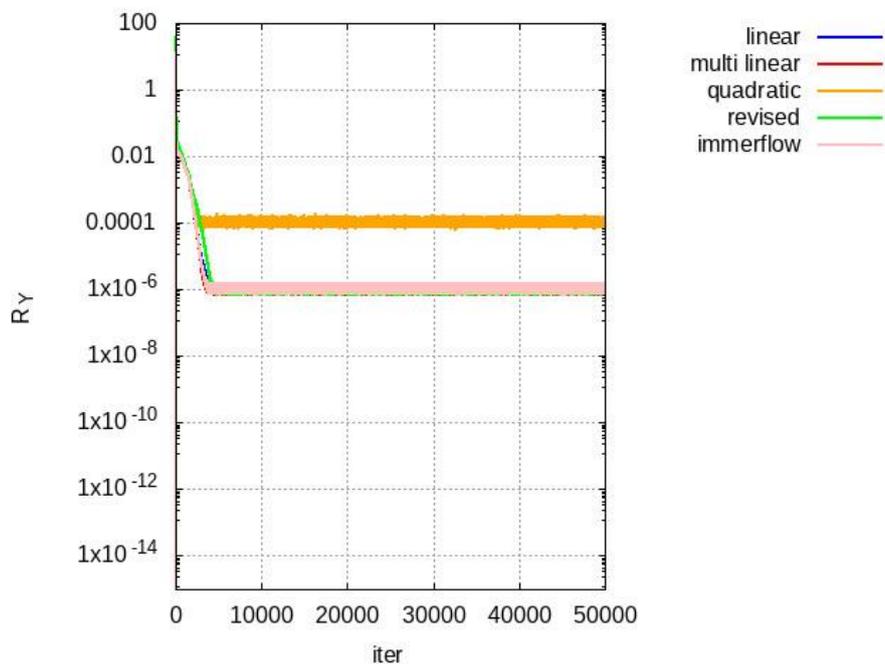


Figura 70: residui dell'equazione di quantità di moto lungo l'asse y.

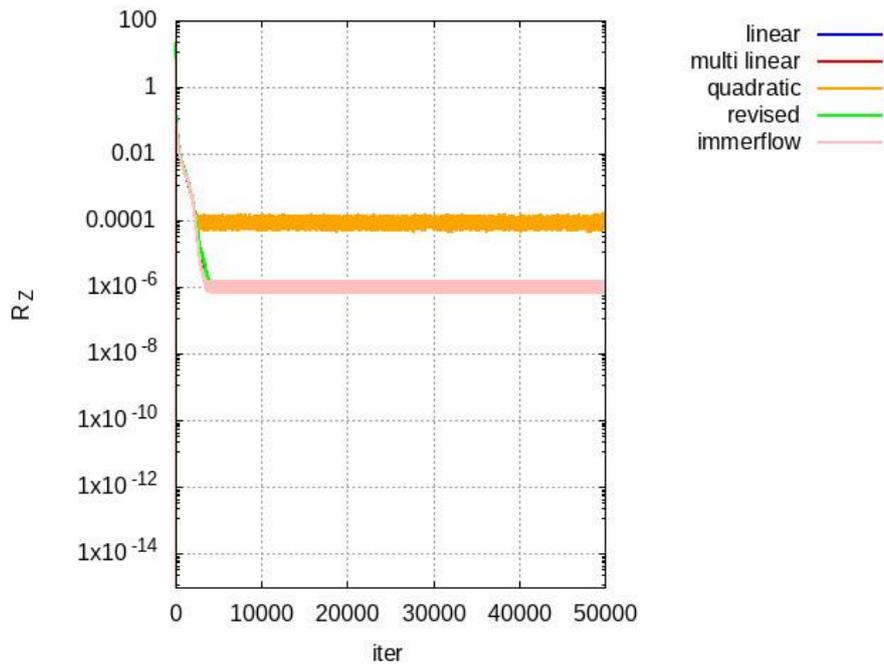


Figura 71: residui dell'equazione di quantità di moto lungo l'asse z.

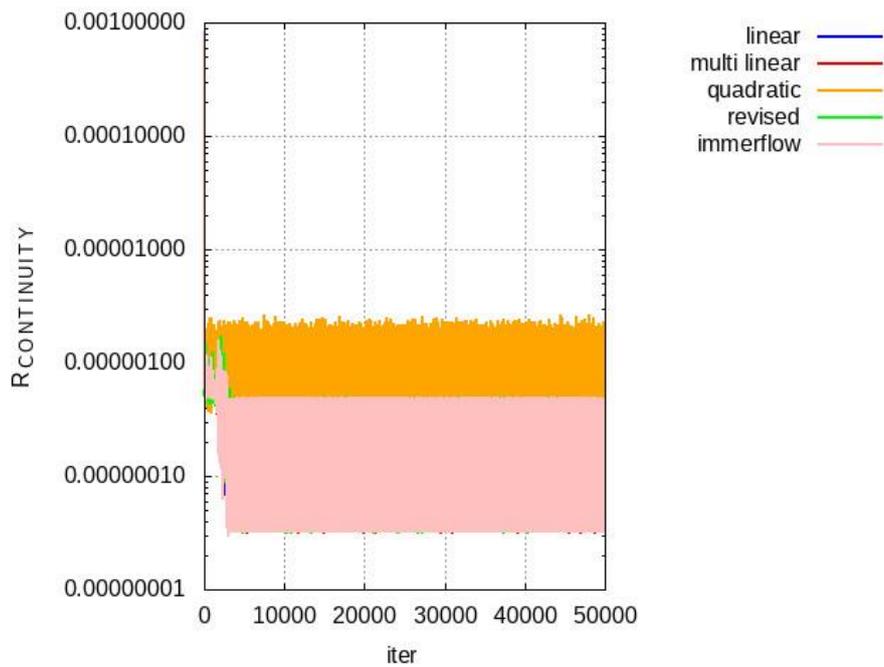


Figura 72: residui dell'equazione di continuità.

Per quanto riguarda le immagini riportanti i residui dell'equazione di continuità si specifica che, il software forza globalmente la soluzione al rispetto della conservazione della massa ad ogni istante temporale, indipendentemente dalla soluzione ottenuta. Questo giustifica i valori bassi ottenuti nelle immagini raffiguranti tali residui. Le oscillazioni sono dovute a rumore numerico creato dalla soluzione numerica del campo.

Capitolo 7

Conclusioni

I risultati ottenuti durante lo svolgimento di questa tesi non si discostano molto da quelli ottenuti con il software *ImmerFlow*, simboleggiando quindi la bontà del codice implementato.

Inoltre, tutti gli schemi implementati presentano buoni risultati in accordo con i risultati analitici e sperimentali con cui si è svolto il confronto, simboleggiando ulteriormente la bontà del software.

Confrontando i risultati ottenuti tra i vari schemi di interpolazione si può notare che lo schema lineare, [paragrafo 5.1](#), a fronte di una minore complessità di calcolo presenta dei buonissimi risultati, aumentando la precisione della soluzione sull'interfaccia *staircase* solido-fluido e mantenendo un costo computazionale basso. Questo è evidenziato soprattutto a confronto con lo schema quadratico, [paragrafo 5.4](#), nel quale il costo computazionale aumenta, senza evidenziare risultati più accurati.

Lo schema bilineare con correzione dovuta al gradiente di pressione, [paragrafo 5.3](#), non porta miglioramenti rispetto alla versione senza correzione, [paragrafo 5.2](#), tra i due è quindi preferibile il secondo che risulti essere di più facile manutenibilità. Non sono state riscontrate neanche le oscillazioni presentate in letteratura, confermando ulteriormente quanto appena descritto.

Resta da migliorare, a livello globale, la parte diffusiva degli schemi, in quanto c'è leggera sovrastima dei gradienti di velocità nella prossimità del solido, come si evidenzia per i casi di lamina piana del [paragrafo 6.2](#), ma, nonostante questo, la soluzione è molto accurata rispetto a quella analitica.

I risultati mostrati sono promettenti ed incoraggiano l'estensione degli schemi, qui implementati, anche per casi turbolenti. Infatti, ci si aspetta che in tali casi possano aiutare a migliorare decisamente la soluzione nei pressi della parete, aumentandone la precisione e il calcolo dei gradienti delle grandezze fluide, molto significativi in questo regime di moto.

Bibliografia & Sitografia

- [1] Fadlun, Verzicco, Orlandi, Mohd-Yusof, “*Combined Immersed Boundary Finite Difference Methods for Three-Dimensional Complex Flow Simulation*”, Journal of Computational Physics 161, 35–60, 2000.
- [2] Mittal, Iaccarino, “*Immersed Boundary Methods*”, Annual Review of Fluid Mechanics 37, 239-261, 2005.
- [3] Kang, Iaccarino, Moin, “*Accurate Immersed Boundary Reconstruction for Viscous Flow Simulations*”, AIAA Journal vol. 47 n.7, 1750-1760, 2009.
- [4] Samouchos, “*The Cut-Cell Method for the Prediction of a 2D/3D Flows in Complex Geometries and the Adjoint-Based Shape Optimization*”, phd thesis, 2022.
- [5] Berger, Olinger, “*Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*”, Journal of Computational Physics, 484-512, 1984.
- [6] Capizzano, “*Turbulent Wall Model for Immersed Boundary Methods*”, AIAA Journal vol. 49 n. 11, 2367-2381, 2011.
- [7] O’Rourke, “*Computational Geometry in C*”, Cambridge University Press, 1998.
- [8] Yang, Balaras, “*An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries*”, Journal of Computational Physics 215, 12-40, 2005.
- [9] Cristallo, Verzicco, “*Combined Immersed Boundary/Large-Eddy-Simulations of Incompressible Three-Dimensional Complex Flows*”, Flow Turbulence Combust 77, 3-26, 2006.
- [10] Palma, de Tullio, Pascazio, Napolitano, “*An immersed-boundary method for compressible viscous flows*”, Computer & Fluids 35, 693-702, 2006.
- [11] de Tullio, Cristallo, Balaras, Pascazio, De Palma, Iaccarino, Napolitano, Verzicco, “*Recent Advances in the Immersed Boundary Method*”, European Conference on Computational Fluid Dynamics ECCOMAS CFD, TU Delft, The Netherlands, 2006.
- [12] Vanella, Balaras, “*A moving-least-squares reconstruction for embedded-boundary Formulations*”, Journal of Computational Physics 228, 6617-6628, 2009.
- [13] Vanella, Rabenold, Balaras, “*A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid–structure interaction problems*”, Journal of Computational Physics 229, 6427-6449, 2010.
- [14] Wang, Yana, Tianb, “*A simple and efficient implicit direct forcing immersed boundary model for simulations of complex flow*”, Applied Mathematical Modeling 43, 287-305, 2016.
- [15] Posa, Balaras, “*Large-Eddy Simulations of a notional submarine in towed and self-propelled configurations*”, Computers and Fluids 165, 116-126, 2018.
- [16] Kalitzin, Iaccarino, “*Turbulence modeling in an immersed-boundary RANS method*”, Center for Turbulence Research, Annual Research Brief, 415-426, 2002.
- [17] Iaccarino, “*The Immersed Boundary Method*”, Mechanical Engineering Department, Institute for Computational Mathematical Engineering, Stanford University, Notes prepared for a Short

Course offered at the University of Bordeaux and INRIA in January 2016, in the framework of the Aquarius project in collaboration between INRIA Bordeaux and Stanford University, 2016.

- [18] Verzicco, “*Large Eddy Simulation Theory and Applications: Immersed Boundary Technique for Large-Eddy-Simulation*”, von Karman Institute for Fluid Dynamics, Waterlooosteenweg 72 – 1640 Sint-Genesius-Rode, 2020.
- [19] Balaras, “*Large Eddy Simulation Theory and Applications: Applications of Large-Eddy and Direct Simulation to Biological Flows*”, von Karman Institute for Fluid Dynamics, Waterlooosteenweg 72 – 1640 Sint-Genesius-Rode, 2020.
- [20] Gorse, Iollo, Telib, Weynans, “*A simple second order cartesian scheme for compressible Euler flows*”, Rapport de Recherche, 1-23, Inria Bordeaux, 2011.
- [21] Osher, Sethian, “*Fronts Propagating with Curvature Dependent speed: Algorithms Based on Hamilton-Jacobi formulation*”, Journal of Computational Physics 79, 12-49, 1988.
- [22] Sethian, “*Advancing Interfaces: Level Set and Fast Marching methods*”, paper, 1999.
- [23] Sapiro, “*Digital image processing: p052 - Level Sets and Curve Evolution*”, video lezione, <https://www.youtube.com/watch?v=1ZJ88JyLPZI>.
- [24] Canuto, Tabacco, “*Analisi Matematica I*”, Springer, 2008.
- [25] Canuto, Tabacco “*Analisi Matematica II*”, Springer, 2014.
- [26] Fey, Koning, Eckelmann, “*A New Strouhal-Reynolds-Number Relationship for Circular Cylinder in the Range $47 < Re < 2 \times 10^5$* ”, Physics of Fluids, 10(7):1547–1549, 1998.
- [27] Nishioka, Sato, “*Measurements of Velocity Distributions in the Wake of a Circular Cylinder at Low Reynolds Numbers*”, Journal of Fluid Mechanics, 65(1):97–112, 1974.
- [28] Paraview documentation, <https://docs.paraview.org/en/latest/>.
- [29] Guida C++, <https://cplusplus.com/>.
- [30] Guida C++, <https://www.html.it/guide/guida-c2/>.
- [31] Guida git, <https://www.atlassian.com/git/tutorials>.
- [32] Williams, Kelly, “*Gnuplot 5.4 An Interactive Plotting Program*”, manual, <http://www.gnuplot.info/documentation.html>.
- [33] Taylor, Whitelaw, Yianneskis. “*Developing Flow in S-shaped Ducts. 1: Square Cross-Section Duct*”, Final Report Imperial Coll. of Science and Technology, 1982.
- [34] Berrone, Pieraccini, Appunti del corso di Metodi Numerici e Calcolo Scientifico, 2020.
- [35] Balaras, “*Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations*”, Computers & Fluids 33, 375–404, 2004.

Appendici

Appendice A: Staggered Grid

A seconda di dove vengono salvate le informazioni su di una griglia si possono avere due tipi mesh:

1. Collocated grid.
2. Staggered grid.

Per una griglia di tipo *collocated*, le informazioni di tutte le variabili del campo di moto fluido, per le celle del dominio, sono salvate nella stessa posizione che usualmente coincide con il centro del volume contenuto nella cella.

Al contrario per una griglia di tipo *staggered*, le variabili incognite del problema sono salvate in posizioni diverse della cella.

Le variabili scalari (come densità, pressione, entalpia, etc.) sono salvate nel centro del volume, mentre variabili vettoriali (velocità, vorticità) sono posizionate sulle superfici di contorno della cella. Spesso per i metodi *Immersed Boundary* si assiste anche alla separazione delle componenti dei vettori: assegnando ad un'interfaccia "x" la componente lungo tale asse del vettore e così anche per le altre due componenti cartesiane.

Utilizzando una griglia *staggered* è più facile evitare problemi di disaccoppiamento tra il campo di velocità e il campo di pressione, ma per ogni variabile bisogna definire differenti domini volumi di controllo da mantenere per tutto il continuo del calcolo.

Nella figura A.68 si può vedere un esempio di griglia *staggered*, sono evidenziate le posizioni in cui vengono salvate le diverse variabili nel caso di campo di moto incompressibile e bidimensionale, si può vedere anche con differenti tratteggi i volumi di controllo delle stesse grandezze del campo.

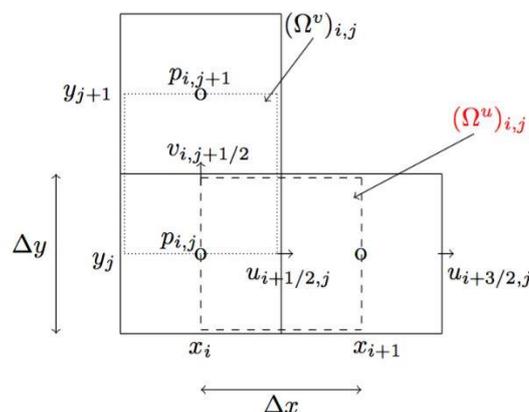


Figura A.73: griglia staggered.

Un esempio sull'utilizzo delle griglie *staggered* è riportato in [\[8\]](#).