





Progetto MIUR DM 593 no. 12656 decreto dirigenziale n. 2266 del 29/12/2003

Vento di Sardegna

Obiettivo Realizzativo n. 1

ORGANIZZAZIONE

Attività 1.3A1

Messa a punto e integrazione dei diversi software per la simulazione

Sottoattività: sviluppo della tecnica "immersed Boundary" parte prima: curve chiuse bidimensionali

partner responsabile

CRS4

Autori: G. Delussu, M. Mulas e M. Talice (CRS4) 28 Febbraio 2005

Capitolo 1. I Metodi dei contorni immersi

Introduzione

La fluidodinamica computazionale (o CFD, acronimo dall'inglese Computational Fluid Dynamics), nasce intorno agli anni '60 per applicazioni di tipo aeronautico. La potenza dei calcolatori dell'epoca limitava fortemente l'utilizzo della CFD per applicazioni di reale interesse industriale, costringendo gli studiosi ad operare drastiche semplificazioni dei modelli fisico-numerici. A partire da quei primi albori, la crescita esponenziale della potenza di calcolo disponibile ha consentito un decisivo sviluppo tanto dei metodi numerici, quanto della modellistica fisica, sicché la CFD é uscita dal mondo accademico per entrare sempre più prepotentemente in quello industriale.

Essa, infatti, permette l'analisi di fenomeni fisici complessi senza dover ricorrere alla realizzazione di prototipi e alla loro sperimentazione in laboratorio, con una drastica riduzione di tempi e di costi nello sviluppo di nuovi prodotti.

L'utilizzo dei metodi della CFD per lo studio di un generico problema di flusso prevede un iniziale passo di discretizzazione, mediante il quale le equazioni continue che governano il fenomeno vengono trasformate in un insieme discreto di relazioni algebriche, che sono definite nelle locazioni spaziali in cui il dominio di interesse é stato suddiviso (le celle della griglia computazionale). Tanto l'accuratezza della soluzione, quanto le proprietà di convergenza degli algoritmi numerici utilizzati per l'integrazione delle equazioni, dipendono dalla qualità di detta discretizzazione. Una "buona" griglia dovrebbe discostarsi il meno possibile dal caso ideale di griglia Cartesiana regolare, per la quale tutti gli algoritmi numerici sono stati sviluppati. Quando non é importante includere nel modello matematico i fenomeni fisici che possono occorrere all'interno delle strutture solide interagenti con il flusso (quali, ad esempio, le distribuzioni volumetriche di campi di temperatura o di massa nel solido, o l'interazione fluido-struttura), la sola parte dello spazio che deve essere discretizzata é quella occupata dal fluido. In ogni caso, il dominio computazionale é sempre, almeno in parte, confinato dalla presenza di superfici solide, e la griglia computazionale deve seguire la definizione geometrica di dette superfici. Nel caso di contorni geometrici di forma regolare e semplice, il problema della generazione di griglie di buona qualità é stato risolto e oggi esistono molti ottimi programmi commerciali dedicati alla generazione delle griglie. Nel caso di geometrie complesse, la generazione delle griglie é ancora un problema importante, a volte d'impossibile soluzione. Si tenga presente che nel caso di geometrie complesse, la generazione della griglia occupa da sola mediamente più della metà del tempo totale del processo di simulazione, tanto da far perdere alla CFD molti dei suoi vantaggi sulla pratica sperimentale quando si dovessero generare molte griglie su geometrie differenti, come nel caso di una normale procedura di progettazione.

Per questo motivo, negli ultimi anni é cresciuto l'interesse per metodi numerici alternativi, che riescano al contempo ad eliminare il problema della generazione delle griglie ed a fornire eccellenti doti d'accuratezza ed efficienza computazionale. Nei capitoli che seguono sarà fornito un esame bibliografico dei metodi proposti in letteratura, ponendo particolare accento su quelli più promettenti per le applicazioni industriali.

Analisi bibliografica

Si é visto nell'introduzione, che nel caso di geometrie complesse la generazione di griglie che seguano il contorno delle superfici solide (Bodyfitted, BF), può risultare così onerosa da rendere impraticabile l'utilizzo della CFD come metodo di progetto. L'idea che sta alla base del metodo dei contorni immersi (o Immersed Boundaries Method, IBM), é dunque quella di generare la griglia di calcolo senza tener conto della reale presenza delle superfici solide, come nell'esempio illustrato nella Figura 1. Il processo di generazione della griglia può così ridursi alla creazione di una griglia Cartesiana ortogonale.



Figura 1: esempio di griglie computazionali per IBM (sinistra) e BF (destra). Nel caso di IBM la presenza del corpo é totalmente trasparente alla griglia, ed é riportata in figura solo per chiarezza. Nel caso di BF la griglia computazionale deve seguire il contorno geometrico del corpo. Le due griglie sono analoghe per il numero di celle lungo il contorno del corpo immerso.

É evidente che anche nel caso di IBM la presenza del corpo solido nel campo di moto del fluido deve in qualche modo essere tenuta nel conto. Poiché questo non avviene durante la fase di discretizzazione spaziale, deve essere fatto durante la fase di soluzione. In altre parole é lo stesso algoritmo numerico che deve contenere in se l'informazione della presenza del corpo nel campo di

moto. In altre parole, si sta spostando la complicazione del problema dalla fase di generazione della griglia a quella della soluzione.

L'idea dell'IBM non é particolarmente nuova, risalendo, infatti, ad alcune decadi or sono, anche se é tornata in auge in anni recenti. In letteratura si trovano molti approcci differenti al problema dell'IBM, e si parla di Immersed Boundaries Methods, Cut Cells Methods, Cartesian Grid Methods, Gridless Methods, Mask Methods, Level-set Methods, Penalty Methods e così via. Il concetto é in ogni modo sempre quello di eliminare la presenza dei corpi solidi dal processo di discretizzazione spaziale, per tener conto della loro presenza nel campo di moto durante la fase di soluzione numerica del sistema di equazioni di Eulero o di Navier-Stokes, a seconda che la viscosità sia introdotta o no nel modello numerico.

Qui di seguito viene riproposto ed integrato l'eccellente esame bibliografico disponibile nel lavoro di Verzicco e Iaccarino [17]. Per ragioni storiche si prenderà dapprima in considerazione la bibliografia sull'IBM, rimandando quella disponibile sugli altri metodi di calcolo sui contorni immersi alla parte finale di questa presentazione. L'idea alla base dell'IBM é, come già ricordato, quella di tener conto della presenza delle superfici solide "immerse" nel dominio computazionale, mediante l'introduzione nelle equazioni che regolano la dinamica dei fluidi, di un opportuno termine forzante.

In termini matematici, nello scrivere le equazioni di Navier-Stokes per fluido incomprimibile, si tiene conto della presenza del contorno immerso mediante l'aggiunta di un opportuno termine forzante, **f**:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = v \nabla^2 \mathbf{u} - \rho^{-1} \nabla p + \mathbf{f}$$
$$\nabla \cdot \mathbf{u} = 0$$

La natura e la forma di \mathbf{f} é stata oggetto del lavoro di diversi autori dalla nascita del metodo fino al giorno d'oggi.

La prima pubblicazione disponibile in letteratura sull'IBM, é quella di Vieceli [1], che sostanzialmente estende il metodo proposto da Harlow e Welch [2,3] (Marker And Cell Method, MAC) al caso di contorni immersi di forma arbitraria. L'idea alla base del lavoro di Viceli consisteva nel considerare la superficie di interfaccia fluido – solido (il contorno immerso, appunto), alla stregua di una superficie libera, e di imporre su di essa condizioni al contorno per la pressione tali che le particelle di fluido potessero muoversi solo lungo la tangente al contorno immerso. Seguendo questo tipo di approccio, é necessario procedere a un aggiustamento iterativo del campo di pressione e di quello di velocità, cosicché, a convergenza, tanto le condizioni di incompressibilità del flusso quanto quelle di impermeabilità del contorno solido, risultano rispettate. Il metodo, noto come ABMAC (Arbitrary Boundary MAC), fu poi generalizzato da Veceli in un

successivo lavoro [4], per tener conto della possibilità di avere superfici mobili. In questo caso, in aggiunta alle condizioni al contorno sulla pressione, é necessario imporre sul contorno anche condizioni per la velocità. Il metodo si rivelò di particolare interesse poiché consentiva il trattamento di superfici il cui moto risultava da una legge imposta o era conseguenza dell'azione esercitata su di esse dal fluido circostante.

Malgrado il lavoro di Veceli costituisce la nascita del metodo dei contorni immersi, la maggior parte dei lavori di letteratura riconoscono in Peskin l'autore che più ha contribuito allo sviluppo e all'applicazione reale del metodo, almeno nei suoi albori. In particolare, nei primi anni '70, Peskin [5,6] sviluppa una metodologia per la simulazione del flusso sanguigno nella valvola mitralica del cuore, sotto l'ipotesi di flusso bidimensionale a numero di Reynolds molto basso. Il problema del flusso sanguigno nei grandi vasi, e nel cuore in particolare, é di difficile soluzione con tecniche di CFD tradizionale (metodi bodyfitted), per il fatto che l'interazione fluido – struttura tra sangue e pareti dei vasi (o del cuore), non può essere trascurata nella risoluzione del problema. Inoltre le superfici solide si deformano, o sotto l'azione del campo pressorio del sangue (come nel caso dei vasi), o per stimolazione elettrica, come nel caso del cuore. Queste esigenze modellistiche portarono Peskin allo sviluppo di un metodo ai contorni immersi, per poter tener conto sia della deformazione delle pareti dei vasi, sia della mobilità del cuore e della valvola mitralica in particolare. In realtà, nei primi lavori di Peskin [5,6], l'elasticità delle superfici solide non era tenuta in conto, mentre era considerato il solo moto della valvola mitralica. La tridimensionalità del flusso e la deformabilità dei vasi, vennero incluse nel modello nei lavori di Peskin e McQueen [7,8]. Nel metodo proposto da Peskin, le equazioni di Navier-Stokes, scritte in forma incomprimibile, sono risolte su una griglia Cartesiana e uniforme, mentre le fibre elastiche della parete del cuore sono immerse nel fluido. Nel modello sono considerate tanto le forze esercitate dal fluido sulle fibre elastiche, tanto l'azione di esse sul fluido stesso. Fluido e fibre, si scambiano cioè delle forze. Allo scopo di conoscere nel tempo la posizione spaziale delle fibre. Peskin fa uso di un sistema di coordinate Lagrangiane, in moto con la locale velocità del fluido e solidale con le fibre stesse. Le informazioni concernenti la posizione delle fibre e le forze che esse esercitano sul fluido circostante, é quindi trasferita alla griglia Cartesiana e uniforme nella quale le equazioni che regolano il moto del fluido sono risolte. Il sistema di riferimento nel mondo del fluido é quello Euleriano, e dunque, il metodo prevede l'uso di un approccio misto Lagrangiano/Euleriano. Con questo modo di procedere, la forza risultante scambiata tra fluido e fibre, si concreta in una serie di funzioni delta definite nelle prime celle fluide esterne al corpo immerso, che, di conseguenza, non può essere rappresentato adeguatamente su una griglia le cui celle hanno dimensioni finite. All'aumentare del numero di celle, si avrebbe ovviamente una riduzione dell'errore, ma a prezzo di un insostenibile costo computazionale. Per questo motivo, Peskin utilizza un metodo per ottenere una transizione più dolce tra il fluido esterno e l'interno del corpo immerso, il che é matematicamente equivalente a rilassare le funzioni delta su una banda di tre o quattro celle computazionali nell'intorno dell'interfaccia fluido – corpo immerso. Il metodo proposto da Peskin é dunque in grado di trattare il caso di interazione fluido – struttura, dove le pareti del corpo solido (nel caso specifico, quelle del cuore), rispondono alle forze su di esse esercitate dal fluido, ed esse stesse esercitano un sistema di forze sul fluido per via della legge di moto imposta. Al di là delle applicazioni di tipo bio-medicale, per le quali il metodo di Peskin fu ideato, ogni applicazione nella quale l'interazione fluido – struttura é parte essenziale del problema, e dove la struttura solida ha eminentemente natura elastica, ben si presta a essere studiata tramite l'applicazione del metodo di Peskin. In realtà lo stesso Peskin ha applicato il metodo ad ambiti differenti da quelli squisitamente bio-medicali, come nel caso dello studio del flusso bidimensionale del moto di un paracadute [43]. Nella maggior parte delle applicazioni di tipo industriale, però, la situazione non é così complicata. Le pareti dei corpi solidi immersi nel fluido possono spesso essere considerate rigide ed indeformabili, e, con una scelta opportuna del sistema di riferimento, anche ferme. Il metodo di Peskin potrebbe in linea di principio essere utilizzato anche in questi casi, riducendo (o annullando) la deformabilità delle fibre. Sfortunatamente, un approccio di questo tipo da luogo a problemi di natura numerica, in quanto la formulazione del problema diviene stiff.

I primi esempi d'applicazione dell'IBM a problemi con contorni immersi solidi e indeformabili, sono quelli dei lavori di Basdevant e Sadourny [9], Briscolini e Santangelo [10] e Goldstein, Handler e Sirovich [11]. Briscolini e Santangelo utilizzarono l'IBM (il metodo prese il nome di mask method, versione modificata del metodo di Basdevant e Sadourny [9]) per simulare il campo di moto bidimensionale non stazionario, intorno a cilindri a sezione circolare e quadrata. Goldstein, Handler e Sirovich [11] nel loro lavoro utilizzarono i contorni immersi per la simulazione del flusso di start up, non stazionario e bidimensionale, intorno ad un cilindro, e per la simulazione del flusso tridimensionale piano e turbolento in un canale a superficie rugosa (ribbed). In questi lavori, l'IBM é usato insieme con un metodo spettrale e il termine forzante é applicato su una banda di tre o quattro nodi computazionali intorno all'interfaccia. L'utilizzo dei metodi spettrali, infatti, può dare luogo alla nascita di modi spuri nella risoluzione delle equazioni nel caso in cui la forzante è applicata sulla sola cella più vicina all'interfaccia, cosicché, allo scopo di limitare l'insorgenza di questo tipo di fenomeni numerici, si rende necessario l'utilizzo di più celle. Saiki e Biringen [12] utilizzarono lo stesso metodo per il calcolo del termine forzante di Goldstein, Handler e Sirovich [11], per il calcolo del flusso intorno a cilindri circolari sia stazionari sia rotanti. Nella loro formulazione, però, si fa uso di un'approssimazione di quarto ordine alle differenze finite, invece che del metodo spettrale. Questo permise di ovviare al problema dell'insorgenza di modi spuri nella soluzione anche nel caso di termine forzante applicato sulle prime celle in prossimità del contorno, utilizzando una procedura simile a quella delle funzioni delta di Peskin [5], procedura che gli autori chiamarono accurata al primo ordine. La maggiore limitazione del metodo proposto da Goldstein, Handler e Sirovich [11], é che nell'espressione della forzante compaiono due termini costanti, del tutto arbitrari, il cui valore deve essere "aggiustato" in funzione dello specifico problema che s'intende risolvere. Nel caso di calcolo non stazionario, poi, il termine forzante introduce una severa limitazione nel passo temporale, che, di fatto, riduce tanto l'efficienza quanto l'effettiva applicabilità del metodo stesso. Infine, un altro limite del metodo proposto in [11], é che, allo scopo di limitare l'insorgere d'oscillazioni spurie della soluzione, il termine forzante deve essere imposto su una fascia di celle intorno all'effettivo contorno fisico, riducendo, di fatto, l'accuratezza con la quale si tiene conto nel calcolo della reale geometria delle superficie solide.

Da un punto di vista matematico, la formulazione formale del problema é dovuta a LaVeque e Calhoum [13], che formularono il problema della posizione dell'interfaccia servendosi di un'equazione di diffusione monodimensionale, non stazionaria. L'equazione proposta in [13] é:

$$\varphi_{t} = \varphi_{xx} + c(t)\delta(x - \alpha(t)) \qquad (1)$$

nella quale $\alpha(t)$ rappresenta la posizione dell'interfaccia (il contorno immerso), c(t) il valore della soluzione in $\alpha(t)$, e i pedici t e x indicano derivazione rispetto allo spazio e tempo (il doppio pedice indica la derivata seconda). La risoluzione numerica della (1) richiede l'utilizzo di un'approssimazione δ_h della funzione forzante δ (essendo h il passo della discretizzazione spaziale utilizzata). LaVeque e Calhoum dimostrarono che sotto opportuni vincoli per δ_h é possibile ottenere formale accuratezza spaziale di secondo ordine sul contorno immerso. Come spesso accade però, i risultati ottenuti nel semplice caso monodimensionale non possono essere direttamente estrapolati al caso bi o tridimensionale, cosicché non esiste una "ricetta" a priori per la forma che deve avere la funzione dh nel caso multidimensionale. Ciò nonostante lavori di letteratura come quello di Lai e Peskin [14], dimostrarono che é possibile conseguire formale accuratezza di secondo ordine anche nel caso tridimensionale. Da quanto finora visto, il principale problema dell'introduzione di termini forzanti, risiede nella nascita d'oscillazioni spurie nella soluzione delle equazioni, che possono inficiare la stabilità numerica del metodo. Si é anche visto che un metodo per ovviare o limitare questo problema é quello di "sparpagliare" il termine forzante su una banda di celle nell'intorno della posizione dell'interfaccia. Il che però riduce l'accuratezza della rappresentazione geometrica del contorno immerso. Mohd-Yusof [15] riuscì a derivare una formulazione alternativa alla forzante che non solo non compromette la stabilità numerica del metodo, ma non richiede rilassamento alcuno della stessa forzante. Per di più, contrariamente a ciò visto per il metodo proposto in [11],

non viene fatto uso d'alcuna costante arbitraria, rendendo il metodo del tutto indipendente dal particolare problema da risolvere.

Mohd-Yusof dimostrò la forza della nuova metodologia in [15], dove la nuova espressione del termine forzante era combinata con l'uso di B-splines per la rappresentazione geometrica dei contorni immersi. Il metodo fu usato con successo per la simulazione del flusso in un canale tridimensionale a superficie rugosa (ribbed channel), fornendo risultati migliori di quelli ottenibili con le formulazioni precedenti. Mentre l'originaria formulazione di Mohd-Yusof fu sviluppata in congiunzione con un metodo spettrale, in tempi più recenti Iaccarino e Verzicco [17] e Kalizin e Iaccarino [18], hanno fatto uso della formulazione di Mohd-Yusof per il termine forzante in un codice LES (Large Eddy Simulations) alle differenze finite. Il codice é stato utilizzato con successo su un vasto intervallo di numeri di Reynolds. Nelle stesse referenze [17] e [18], sono riportati alcuni dei risultati ottenuti.

Come detto, la formulazione IBM che fa uso del termine forzante, é solo uno dei possibili modi mediante cui é possibile dare soluzione al problema di includere un contorno "immerso" nel dominio computazionale. Tra gli altri possibili approcci al problema che si trovano in letteratura, vi é il cosiddetto penalty method (anche noto come fictious domain method o domain embedding method). Secondo questa metodologia, il corpo immerso é considerato come un corpo poroso e per esso vengono risolte le equazioni di Navier-Stokes-Brinkman, che altro non sono che le solite equazioni di Navier-Stokes con l'aggiunta di un termine di resistenza volumica, detto Darcy drag. Tale termine é quello che descrive l'azione esercitata dal mezzo poroso sul flusso. Una dettagliata e puntuale descrizione del metodo può essere trovata nei lavori di Khandra e al. [21], Angot e al. [22], Kevlahan e Ghidaglia [23]. Nei paragrafi che seguono sarà in ogni caso dimostrato che, con opportuni arrangiamenti, il termine forzante utilizzato nel penalty method può essere facilmente interpretato come una riformulazione (di significato fisico differente) dell'IBM proposto in [11] e [12].

I metodi esaminati fino a questo punto, modificano le originarie equazioni di Navier-Stokes mediante l'aggiunta di un termine forzante. Vi é un'altra classe di metodi per trattare il problema dei contorni immersi, che utilizza un approccio diverso. Questi metodi vanno sotto il nome di Cartesian Grid Methods o Cut Cells Methods o ancora Embedded Cells Methods. Secondo quest'approccio, piuttosto che modificare la formulazione delle equazioni di Navier-Stokes tramite l'aggiunta di un termine forzante, si modificano le celle della griglia cartesiana in corrispondenza delle loro intersezioni con il contorno del corpo immerso. Dato l'elevato numero di possibili configurazioni secondo le quali celle della griglia e contorno immerso si possono intersecare, é necessario che l'algoritmo numerico sia capace di calcolare esattamente tutti i parametri geometrici

d'interesse risultanti da dette intersezioni. Malgrado queste complicazioni di natura geometrica, questo tipo di metodologia ben si presta ad essere utilizzata in una formulazione ai volumi finiti, ed infatti eccellenti esempi d'applicazioni a complessi flussi tridimensionali possono trovarsi, ad esempio, in Berger e Aftosmis [24], in Verzicco e al. [41] o in Kim e al. [30]. Nel metodo dei volumi finiti, la risoluzione delle equazioni di Navier-Stokes, richiede il calcolo del flusso delle variabili indipendenti attraverso le facce del volume d'ogni cella della discretizzazione. Infatti, nell'approccio ai volumi finiti, le leggi di conservazione della massa, della quantità di moto e dell'energia, devono essere soddisfatte per ciascuno dei volumi elementari nei quali l'intero dominio di calcolo é suddiviso. Il calcolo dei flussi é ovviamente semplice nel caso di griglie cartesiane e ortogonali, come quelle usate come background nei cut cells methods, in tutte le zone del dominio di calcolo nelle quali non si abbia la presenza del contorno immerso. Laddove però, il contorno del corpo immerso taglia le celle della griglia cartesiana, poiché la tipologia di tali tagli può essere qualsivoglia, non é in generale facile calcolare il flusso attraverso le superfici tagliate, e devono essere sviluppate delle metodologie ad hoc per risolvere il problema. Esempi di come questo problema possa essere affrontato e risolto si possono trovare nei lavori di Aftosmis [24] e di Forrer [49]. Un altro inconveniente che i cut cells methods portano sempre con sé, é quello legato alla formazione di celle particolarmente piccole (rispetto alla dimensione media delle celle della griglia), nelle zone in cui il contorno immerso taglia la griglia cartesiana soggiacente. Conseguenza di ciò, é la nascita di problemi di stiffness in fase di soluzione, che possono diventare così severi da costringere ad utilizzare un passo temporale inaccettabilmente piccolo per l'avanzamento della soluzione, sì da rendere l'algoritmo inutilizzabile. Soluzioni a questo problema sono state studiate e proposte da vari autori. Tra gli altri si possono ricordare qui quelle proposte da Forrer [50], e, più recentemente, da Ye e al [25]. Questo ultimo autore, in particolare, propose una tecnica che fonde le celle più piccole con quelle immediatamente vicine. Tale schema, se da una parte da soluzione al problema della stiffness, dall'altra porta ad una perdita d'accuratezza nella rappresentazione della reale geometria del corpo immerso.

Infine, merita una menzione particolare il metodo proposto da Dadone e al. [48] per la risoluzione delle equazioni di Eulero. La formulazione del metodo, che pur appartiene alla classe dei cut cells methods, é, infatti, tale da eliminare il problema delle celle piccole. Inoltre, in linea di principio, il metodo é estendibile anche al caso viscoso, non presentando differenze particolari nel trattamento delle condizioni di parete rispetto alle altre formulazioni fin qui esaminate.

Un metodo alternativo é quello rappresentato dall'utilizzo di un metodo puramente "gridless", per l'intero dominio computazionale. A questo riguardo si vedano per esempio i lavori di Batina [51] e di Subrata e Fleming [52].Questi metodi fanno di solito uso o di una interpolazione diretta delle variabili di flusso con un metodo ai minimi quadrati (come in [51]), oppure usano interpolazioni mobili ai minimi quadrati delle funzioni test in una formulazione agli elementi finiti, [52]. Mentre metodi di questo tipo erano già stati utilizzati per lo studio della propagazione delle fratture nella meccanica strutturale [53], il primo esempio d'applicazione a problemi di fluido dinamica si deve allo stesso Batina [51]. Questi metodi però, malgrado la loro potenziale flessibilità, hanno lo svantaggio di non garantire l'esatta conservazione delle variabili di flusso. Inoltre, risulta difficile per questa classe di metodi, l'implementazione di algoritmi per l'accelerazione della convergenza, come le tecniche multi-grid. Infine, si ha una generale perdita d'efficienza complessiva del metodo nel caso di griglie strutturate, per il lavoro aggiuntivo che é necessario svolgere per il calcolo delle funzioni interpolanti ai minimi quadrati. Un metodo misto é quello proposto da Koh e al. [54] e da Kirshman e Liu [55], i quali propongono di utilizzare un metodo puramente cartesiano in tutte le zone del dominio computazionale non direttamente interessate dalla presenza del corpo immerso, per riservare l'applicazione delle tecniche gridless alla parte del dominio in vicinanza del corpo. In questo modo si recupera l'efficienza computazionale dei metodi cartesiani quasi ovunque, meno che nella zona dove il corpo é immerso, dove, a scapito di una minore efficienza, si ha una maggiore flessibilità del metodo. Alcuni esempi d'applicazioni si possono trovare nello stesso lavoro di Koh e al. [54].

Un differente tipo d'applicazioni che ha suggerito lo sviluppo di una metodologia in linea di principio applicabile anche al caso più generale dei contorni immersi, é quello dello studio delle superfici libere, o, più in generale, dello studio di un'interfaccia mobile in una griglia fissa. Tra le varie tecniche proposte per la risoluzione di questo tipo di problemi, si possono ricordare la VOF (Volume of Fluid) e il Level Set method. Utili riferimenti bibliografici possono essere trovati nei lavori di Hirt e Nichols [28], per il VOF method, e di Sethian [29], per il Level Set method.

Conclusioni

Nei capitoli che seguono l'attenzione sarà posta sullo sviluppo e l'implementazione di un cut cells method che sia applicabile al particolare problema della simulazione del flusso intorno alle vele. Il

capitolo 2 é un capitolo introduttivo al solutore in-house che é stato utilizzato come base per l'implementazione dei metodi in 2d. Nel capitolo 3 sarà presentato un primo metodo, molto semplice da implementare, in particolare per la parte riguardante il solutore, e ne saranno descritte le varianti nel tentativo di migliorarne i risultati. Nei successivi due capitoli saranno presentati in dettaglio i metodi proposti da Forrer [49] [50] e da Dadone [48], e ne saranno discusse le problematiche d'implementazione. Ogni capitolo è ulteriormente suddiviso in tre parti: la prima parte è dedicata al preprocessore ovvero al calcolo di tutte quelle quantità relative alla griglia ed alla geometria che sono necessarie al solutore e che sono indipendenti dalla soluzione; la seconda parte é incentrata sul solutore e sulle routine che é stato necessario aggiungere o modificare rispetto al solutore iniziale trattato nel capitolo 2; la terza parte riguarda i risultati e l'analisi critica degli stessi.

Capitolo 2 II solutore base

Introduzione

Questo capitolo vuole dare le caratteristiche principali del solutore 2d in-house utilizzato come base per l'implementazione dei vari metodi a contorni immersi. Tale solutore é denominato tharros.

Caratteristiche del Solutore

Il moto di un fluido comprimibile non viscoso e adiabatico nello spazio a due dimensioni é descritto dalle equazioni di Eulero, che nella formulazione conservativa ovvero in forma di flussi sono date da:

$$Qt + Fx + Gy = 0$$

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho e + p) \end{pmatrix}, G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho e + p) \end{pmatrix}$$
(2)

in cui il pedice "t" indica una derivazione rispetto al tempo, i pedici "x" e "y" indicano una derivazione rispetto alle due coordinate principali; "F" e "G" sono i flussi delle variabili conservative "Q" rispettivamente lungo x e lungo y; " ρ " é la densità, "(u,v)" le componenti del vettore velocità, "e" é l'energia per unità di densità e "p" é la pressione.

Con l'assunzione dell'ipotesi di gas perfetto si ha l'ulteriore relazione:

$$p = (\gamma - 1)(\rho e - \frac{1}{2}\rho(u^2 + v^2))$$

É un sistema d'equazioni iperboliche non lineari. Tale sistema è discretizzato nel tempo e nello spazio. Per l'avanzamento nel tempo si é scelto un metodo Runge-Kutta a quattro stadi. Per la cosiddetta semidiscretizzazione nello spazio si é scelto il metodo dei volumi finiti che si adatta naturalmente alla formulazione in forma di flussi. Integrando la (2) sulla superficie di una cella:

$$\frac{\partial}{\partial t} \int_{\Delta S} Q \cdot dS + \int_{\Delta S} (Fx + Gy) \cdot dS = 0$$

che, utilizzando il teorema di Stokes, diventa:

$$\frac{\partial}{\partial t} \int_{\Delta S} Q \cdot dV + \oint_{l} (F\vec{i} + G\vec{j}) \cdot \vec{n} dl = 0$$

con "l" linea che racchiude la superficie, sistema che discretizzata nello spazio diventa:

$$\Delta S \frac{\partial}{\partial t} Q_{i,j} + \sum_{nlati}^{k=1} T n_k \Delta l_k = 0$$

dove:

$$\Delta S \cdot Q_{i,j} = \int_{\Delta S} Q \cdot dS$$
$$Tn = (F\vec{i} + G\vec{j}) \cdot \vec{n}$$

ovvero $Q_{i,j}$ é una media della variabile Q sulla superficie della cella della e Tn é il flusso proiettato lungo la normale al lato "l", flusso calcolato quindi sulle interfacce tra le celle. Per calcolare i flussi sono stati implementati tre schemi del tipo TVD, che costituiscono un solutore approssimato del problema di Riemann, che data la soluzione a destra e sinistra dell'interfaccia trova il valore nell'interfaccia. Gli schemi implementati sono primo ordine, secondo ordine simmetrico e secondo ordine upwind. Il flusso viene dato da una parte centrale e dal contributo TVD, parte dissipativa che tiene conto delle caratteristiche e quindi delle velocità a cui si muovono le informazioni nel flusso i.e. in due dimensioni Vn,Vn+c,Vn-c con Vn velocità normale al lato e c velocità locale del suono.

Capitolo 3. Un Metodo Semplice

Introduzione

Nel corso di questo capitolo sarà trattato un metodo, che per le sue caratteristiche di semplicità della parte solutore, é stato considerato come primo candidato nella simulazione delle vele con i contorni immersi. Il metodo prevede un trattamento speciale solo per le celle tagliate dalla boundary mentre le rimanenti celle sono trattate nel consueto modo. In realtà anche se le celle interne sono calcolate come tutte le altre, escluse come detto quelle di boundary, tutti quei contributi, che dovrebbero andare a cambiare i residui delle variabili conservative e quindi in fase di update, o aggiornamento, le variabili stesse, sono annullati. Il campo di moto é quindi bloccato alle condizioni iniziali per le celle interne, calcolato nel modo classico per le celle esterne e calcolato in modo speciale per quelle di boundary. In particolare ciò che è calcolato sono i flussi attraverso le boundary e nel caso delle celle di boundary viene aggiunto un contributo di pressione. Maggiori dettagli saranno dati nella sezione concernente il solutore.

Preprocessore

Nell'ambito del presente metodo, ma in generale in tutti i metodi a contorni immersi o celle tagliate, é necessaria la presenza di un preprocessore che analizzi la geometria e fornisca al solutore alcune variabili che permettano di tenere conto della presenza della geometria stessa nei calcoli.

I file richiesti in input sono la griglia di background, che é la griglia su cui il solutore andrà a risolvere le equazioni della fluidodinamica, e la curva, aperta o chiusa, che si trova all'interno della griglia. Da tastiera invece sono richiesti il grado della spline, l'eventuale cambio della direzione della normale, il blocco di una o più direzioni di sweep e la tolleranza.

In output il preprocessore fornisce una lista delle celle della griglia di background tagliate dalla curva, la superficie non immersa d'ogni cella, quindi zero per le celle interne alla boundary e pari al volume della cella per le celle esterne, e omegai e omegaj parte di lato, rispettivamente ad i costante e a j costante, non immerso per ogni lato d'ogni cella del dominio fratto la misura dell'intero lato.

Come esempi sono stati scelti metà cilindro, o meglio un semicerchio in due dimensioni, e la metà



di una sezione convergente di un ugello, per poter confrontare i risultati con una soluzione teorica di riferimento nella parte solutore. In questa parte dedicata al preprocessore sì fará riferimento solo al cilindro presentando le due geometrie, dal punto di vista del preprocessore, problematiche simili.

Funzionamento del Preprocessore

In Figura 2 é schematizzato, attraverso un diagramma di flusso, il funzionamento del preprocessore.

Il primo passo é la lettura della griglia di background. Un esempio di griglia di background per il cilindro é dato in Figura 3. Si può notare che si tratta di una semplicissima griglia cartesiana. In seguito è letta la curva, nel nostro caso il semiclindro. Sempre in Figura 3 é mostrato anche il semicilindro. La grandezza della mesh rispetto al cilindro é dovuta alla necessitá di avere flusso imperturbato alle boundary del dominio dove sono imposte le boundary condition di free stream. La curva è data come set di punti. Una volta letta la curva, si passa al controllo delle dimensioni relative della curva rispetto alle dimensioni della griglia di background. Se la curva ha troppi pochi punti si provvede ad aumentarli. Il metodo per aumentarli é far passare una curva di tipo B-Spline attraverso i punti dati e valutarla in un numero di punti superiore a quello iniziale.

Il risultato non perde quindi informazioni geometriche rispetto all'input iniziale.

Un'altra informazione collegata alla curva é la possibilità di bloccare una direzione di sweep. Si parlerà di questo a proposito della routine che trova le celle interne alla boundary.

Quindi si passa alla ricerca delle celle della griglia di background che sono tagliate dalla curva. Senza voler andare troppo a fondo nell'algoritmo si fa un primo loop su tutte le celle per individuare le celle che sono più vicine ad ogni singolo punto della curva; su quelle celle si fa il controllo se il punto appartiene alla cella oppure no; se il punto appartiene si associa il punto alla cella. Questa associazione è indispensabile in seguito. Una volta determinate le celle tagliate si controlla se ci sono dei "buchi" ovvero se nella sequenza mancano delle celle. Questo é determinato dal fatto che alcune celle sono tagliate in piccolissima parte dalla curva e le probabilità che uno dei punti con cui si é definita la curva ricada proprio in quelle celle, é ovviamente ridotta. Una volta accertata la presenza di un buco allora s'infittiscono i punti della curva in quella zona, tramite



valutazioni della B-Spline, e per ogni punto si fa il test d'appartenenza alla cella che riempirebbe il buco.

La ricerca delle normali alla boundary da associare alle celle é effettuata ricordandosi che ad ogni cella é associato un punto della boundary. Basterà quindi valutare la normale alla B-Spline nel punto. Il metodo quindi presuppone una normale per ogni cella o, che é lo stesso, la curva è approssimata in ogni cella da una retta che passa per il punto associato alla cella ed ha la normale alla curva in quel punto.

Il passo successivo é la determinazione delle superfici immerse e degli omega che sono definiti come la parte di lato non immerso sul lato totale. A tal fine si utilizza la semplificazione menzionata in precedenza che la curva può essere approssimata dentro la cella da una retta. Una retta può tagliare o non tagliare una cella in cinque possibili modi, come illustrato in Figura 4. Il caso 1 é di cella completamente esterna alla retta. L'area della superficie non immersa é pari all'area della cella e gli omega dei quattro lati sono tutti pari all'unità. Il caso 2 é di cella completamente interna rispetto alla retta e quindi alla boundary. L'area della superficie non immersa é zero e così anche gli omega dei quattro lati. Nel caso 3 la cella ha un solo vertice non immerso. L'area della superficie non immersi é quella di un triangolo. Per due lati su quattro omega é nullo mentre per gli altri vanno calcolate le intersezioni retta-lato e dedotta la omega. Nel caso 4 due vertici sono non immersi e due immersi. L'area da calcolare é quella di un quadrilatero. Uno degli omega é unitario, uno é nullo e gli altri due sono da calcolare mediante le intersezioni retta-lato. Il caso 5 invece rappresenta la situazione in cui un solo vertice é immerso. L'area della superficie immersa é quella di un poligono a cinque lati e può più agevolmente essere calcolata come differenza tra l'area totale e l'area del triangolo immerso. Due omega sono unitari e due da calcolare con le intersezioni retta-



Figura 4 Modi in cui una retta puo' tagliare una cella di background.

lato.

Nella fase successiva si trovano le celle interne. Le celle sono divise in esterne, interne e di boundary e per fare ciò si eseguono sweep lungo le quattro direzioni +i,-i,+j,-j. In fase di input si può dire di bloccare una o più di queste direzioni e questo serve nel caso ad esempio del semicilindro. La direzione di sweep é una direzione lungo la quale le celle sono considerate. Per



Figura 5 Divisione delle celle della griglia di background per il semicilindro.

esempio se la direzione di sweep é +i per tutte le colonne si parte da i=1 e si procede nella direzione delle i positive fino a che non si trova una cella di boundary. Tutte le celle che si attraversano per arrivare a tale cella sono celle esterne. Per le altre direzioni si procede in modo analogo. In Figura 5 sono mostrate le celle esterne, interne e di boundary per il semicilindro ottenute con il blocco della direzione +j (verticale dal basso verso l'alto).

Solutore

Ricordando la formulazione dei flussi vista nel capitolo precedente:

$$F = \begin{pmatrix} \rho u \\ \rho u^{2} + p \\ \rho u v \\ u(\rho e + p) \end{pmatrix}, G = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^{2} + p \\ v(\rho e + p) \end{pmatrix}$$

si può dedurre che l'unico contributo nel caso di parete solida viene dalla pressione e solo alle due equazioni che esprimono a conservazione della quantità di moto lungo x e y. Quindi l'idea del presente metodo é quella di aggiungere nelle celle tagliate il contributo di pressione dato dalla parete solida e di sommare ad esso i contributi degli altri lati nella misura della parte di lato che rimane al di fuori della boundary ovvero nella misura delle omega (omegai o omegaj a seconda che si tratti di lato ad i o a j costante). Tradotti in formule i due contributi per i residui della quantità di moto lungo x e lungo y per una cella tagliata di indici i e j sono:

$$- p_{i,j}[(1 - \omega i_{i,j})Li_{i,j} - (1 - \omega i_{i+1,j})Li_{i+1,j}] - p_{i,j}[(1 - \omega j_{i,j})Lj_{i,j} - (1 - \omega j_{i,j+1})Lj_{i,j+1}]$$

in cui $p_{i,j}$ é la pressione nella cella (i,j), $(1 - \omega i_{i,j})Li_{i,j}$, $(1 - \omega i_{i+1,j})Li_{i+1,j}$, $(1 - \omega j_{i,j})Lj_{i,j}$, $(1 - \omega j_{i,j+1})Lj_{i,j+1}$ sono le parti di lato immerso rispettivamente per il lato ad i costante, ad i+1 costante, a j costante, a j+1 costante. Nel caso di griglia non allineata con gli assi x e y la formula assume la forma più generale:

$$-p_{i,j}\{[(1-\omega i_{i,j})Lix_{i,j} - (1-\omega i_{i+1,j})Lix_{i+1,j}] + [(1-\omega j_{i,j})Ljx_{i,j} - (1-\omega j_{i,j+1})Ljx_{i,j+1}]\} -p_{i,j}\{[(1-\omega j_{i,j})Ljy_{i,j} - (1-\omega j_{i,j+1})Ljy_{i,j+1}] + [(1-\omega i_{i,j})Liy_{i,j} - (1-\omega i_{i+1,j})Liy_{i+1,j}]\}$$

in cui la lunghezza L ha componenti sia lungo l'asse delle ascisse che lungo quello delle ordinate.



Figura 6 Contributi dovuti alla pressione in una cella di boundary.

In Figura 6 é mostrato un esempio in cui sono chiariti i vari termini che compongono i due contributi alle equazioni della conservazione della quantità di moto dovuti alla pressione nelle celle di boundary ovvero tagliate.

Il contributo di pressione é come detto l'unico per la parte centrale ma occorre aggiungere i termini diffusivi mancanti. Per ottenere il valore delle variabili nelle celle fantasma, necessario al calcolo dei flussi, si impongono le boundary conditions, ovvero le condizioni al contorno, per i lati immersi della boundary. In pratica i calcoli dei contributi diffusivi e del valore delle variabili nelle celle fantasma sono fatti nella stessa routine, giacché il valore delle variabili di ghost non servirà poi a nessun'altra routine. Tutto questo è fatto solo per la parte immersa di ogni lato delle celle di boundary. Per ognuno dei tre schemi é stata aggiunta una routine che calcola il contributo diffusivo. Una volta fatto l'esperimento su diverse geometrie ci si é resi conto che non si ottenevano gli stessi risultati della formulazione per griglia bodyfitted. Per indagare sul fenomeno si é utilizzato una griglia simil-bodyfitted¹ e si é provveduto a controllare lato per lato i vari contributi ai residui delle variabili conservative. Il risultato é che si é riusciti a riprodurre esattamente, a meno di precisione macchina, il comportamento del metodo bodyfitted effettuando alcune aggiunte al metodo appena

¹ Nella sezione relativa ai risultati si chiarirà il significato del termine simil-bodyfitted.

presentato. In particolare si é aggiunta una routine che fornisce la convezione delle componenti u e v della velocità e dell'entalpia, contributo che entra nella parte centrale dei flussi. In formule:

 $(\rho Vn)_L u_L + (\rho Vn)_R u_R$ $(\rho Vn)_L v_L + (\rho Vn)_R v_R$ $(\rho Vn)_L h_L + (\rho Vn)_R h_R$

in cui i pedici L e R indicano rispettivamente a sinistra e a destra dell'interfaccia in cui si stanno calcolando i flussi.

Risultati

Come casi attraverso cui valutare la bontà del metodo si é scelto un semicilindro in basso flusso subsonico e una semisezione lineare di un condotto.

Semicilindro

In Figura 7 é mostrato la griglia bodyfitted relativa al semicilindro di riferimento.



Figura 7 Griglia Bodyfitted di un semiclindro.

I risultati di questa griglia sono stati confrontati con quelli di una griglia del tipo di quella in Figura 3. In particolare si é indagato sull'effetto sulla soluzione dello spostamento relativo tra griglia di



Figura 8 Particolare dell'intersezione tra griglia di background e curva di boundary a y=0.

background e curva di boundary. Per fare ciò, si veda in Figura 8 il lato evidenziato dall'ellisse, si sono costruite undici griglie in cui la prima cella tagliata ad ordinata nulla interseca la curva di boundary per la prima griglia allo zero percento del lato(come in figura), ovvero allo spigolo di sinistra e così via fino all'ultima griglia che interseca la curva al cento percento del lato, ovvero in corrispondenza dello spigolo di destra. I risultati evidenziano differenze piuttosto limitate ma esistenti. Si veda a titolo d'esempio in Figura 9 il campo di pressione per uno schema del primo ordine in vicinanza del semicerchio per una griglia con intersezione allo zero percento ed una con intersezione intorno al quaranta percento.



Figura 9 Campo di pressione per due griglie tagliate dalla boundary in modo differente alla base. Allo 0% quella in alto e al 40% quella in basso.

In Figura 10 é invece mostrato il campo di pressione per uno schema del primo ordine e alcune streamlines della griglia con intersezione al 40% confrontato con quello della griglia bodyfitted. Si può notare che per la griglia a contorni immersi la formulazione correttamente garantisce che il corpo sia riconosciuto e che il flusso gli giri attorno, come nel caso bodyfitted. Il campo di pressione invece rivela una sovrapressione in corrispondenza del punto d'arresto più accentuata

nella griglia bodyfitted così come una depressione più marcata ed un recupero a fine semicilindro che nel caso a contorni immersi manca completamente. Gli stessi risultati si ottengono passando ad uno schema del secondo ordine, si veda Figura 11.



Figura 10 (sopra). Campi di pressione e streamlines per la griglia bodyfitted (nella parte superiore) e per una griglia a contorni immersi al 40% di lato di base(nella parte inferiore). Schema primo ordine.

Figura 11(sotto). Campi di pressione per la griglia bodyfitted (nella parte superiore) e per una griglia a contorni



immersi allo 0% del lato di base(nella parte inferiore). Schema secondo ordine simmetrico.

Per cercare di capire quali elementi mancavano al metodo per ottenere risultati paragonabili al bodyfitted si sono fatte delle prove su una griglia simil-bodyfitted, una griglia in altre parole adattata al corpo e tale che le intersezioni tra curva e griglia fossero sempre le stesse per ogni cella in modo da poter riprodurre esattamente, nel caso in cui l'intersezione coincide con la linea di griglia, la soluzione bodyfitted. La Figura 12 mostra una griglia di tale tipo con intersezione a metà della cella. La griglia ha rispetto alla bodyfitted una linea di griglia in più come si può notare dalla posizione del semicilindro (linea nera).



Figura 12 Dettagli di griglia cosiddetta simil-bodyfitted. In nero é disegnato il semicilindro.

Apportando le modifiche riportate nella sezione precedente si sono riottenuti, per una griglia tagliata allo zero percento, gli stessi risultati, in doppia precisione, di una griglia bodyfitted. Ripassando ad una griglia rettangolare, non simil-bodyfitted, il metodo diventa instabile e diverge in poche iterazioni. Tale instabilità sembra essere legata alla presenza di volumi non immersi piccoli rispetto ai volumi circostanti ed in letteratura é un problema già conosciuto. Nei prossimi due capitoli si tratteranno due metodi che con due approcci diversi cercano di aggirare questo problema d'instabilità numerica.

Condotto

In Figura 13 é mostrata la griglia bodyfitted sulla sezione convergente di un ugello. Per simmetria si lavora solo su metà sezione. Come nel caso precedente si sono comparati i risultati ottenuti sulla



Figura 13 Griglia bodyfitted del convergente di un ugello.

griglia bodyfitted con quelli su una serie di griglie, per il metodo dei contorni immersi, che differivano per il punto in cui tagliavano le celle. Esaminando due di queste griglie la traccia del convergente è semplicemente traslata. In Figura 14 é mostrata una griglia di questo tipo. Per una griglia differente la curva in nero é traslata verso il basso o verso l'alto. Sono state costruite 7 griglie che tagliano in modo differente la cella 49 all'inlet (sinistra in figura) e di conseguenza anche le celle della parte convergente.



Figura 14 Griglia a contorni immersi del convergente di un ugello.

Come condizioni al contorno si assegnano la pressione totale e la temperatura totale all'inlet (sinistra in figura) e la pressione all'outlet (destra in figura). Ne derivano univocamente i valori di tutte le grandezze fluidodinamiche all'inlet e all'outlet. I risultati mediati, all'inlet e all'outlet, tra le varie griglie ai contorni immersi variano, tra loro, dell'ordine del 5% nella componente verticale della velocità e sotto al punto percentuale per la pressione, densità, temperatura e componente orizzontale della velocità. Il confronto rispetto al metodo bodyfitted ed ai risultati teorici é invece dato in tabella. 1. I risultati teorici sono stati ottenuti con la semplificazione di quasi-1d.

Tabella 1Confronto dei risultati delle griglie bodyfitted e a contorni immersi con i risultati dedotti dalla teoria.

Inlet					
	Teorico	Bodyfitted	Contorni	Bodyfitted	Contorni
		primo ordine	immersi primo	secondo ordine	immersi
			ordine	simmetrico	secondo ordine
					simmetrico
Pressione [Pa]	106303	106431	106412	106359	106358
Velocità	49.31	47.21	47.52	48.41	48.42
orizz[m/s]					
Mach	0.144	0.138	0.139	0.141	0.141
Densità[Kg/m3]	1.269	1.270	1.270	1.269	1.269
Temperatura[°C]	292.0	292.1	292.1	292.0	292.0
Outlet					
Pressione [Pa]	101325	101596	101553	101510	101510
Velocità	102.05	97.25	97.97	99.50	99.61
orizz[m/s]					
Mach	0.3	0.288	0.290	0.295	0.295
Densità[Kg/m3]	1.226	1.227	1.227	1.228	1.228
Temperatura[°C]	288.0	288.6	288.4	287.9	287.9

Si può notare come i risultati delle griglie ai contorni immersi siano in ottimo accordo con quelli teorici e quelli della griglia bodyfitted. Analizzando oltre ai risultati mediati il campo delle variabili fluidodinamiche, si veda in Figura 15 l'andamento della pressione, si nota invece una peggiore definizione delle curve di livello nelle griglie ai contorni immersi e la comparsa di alcuni valori errati in corrispondenza di celle con un volume non immerso piccolo (celle blu verso alla fine del convergente).



Figura 15 Andamento della pressione nel convergente di un ugello per la griglia bodyfitted (alto) ed ai contorni immersi (basso). Schema simmetrico secondo ordine.

Riassumendo i risultati sono buoni per la geometria convergente di un ugello ma assolutamente insoddisfacenti per il cilindro. Ciò ha portato a considerare altri metodi per il metodo ai contorni immersi.

Capitolo 4. Il metodo di Forrer

Introduzione

Si tratta di un metodo a griglia cartesiana, che prevede un trattamento speciale solo per le celle di boundary. Il problema principale in questo tipo di metodi é, come si é visto anche nel precedente capitolo, l'instabilità che nasce nel caso di celle di boundary con volumi piccoli, rispetto alle celle vicine, e la necessità di avere un elevato ordine d'accuratezza in corrispondenza delle boundary. Questo metodo dovrebbe raggiungere entrambi gli obiettivi mantenendo caratteristiche che lo rendono facilmente estendibile alle tre dimensioni.

Preprocessore e suo funzionamento

In Figura 16 é mostrato il diagramma a blocchi del preprocessore per il metodo di Forrer. Tutte le considerazioni fatte per il precedente preprocessore sono valide anche per il presente preprocessore



e quindi ci si limiterà a considerare solo la parte che si aggiunge ad esso. Infatti confrontando il diagramma a blocchi con quello del metodo "semplice", trattato nel capitolo precedente, si può notare che l'unica differenza é la presenza di una routine supplementare detta di ricerca dei vicini delle celle riflesse. Si tratta di una routine che cerca i tre centri cella più vicini ai punti ottenuti dal ribaltamento o riflessione dei centri cella delle prime due celle interne rispetto alla curva di boundary, o meglio rispetto all'approssimazione lineare di tale curva. Le celle interne sono quelle celle che confinano con una cella tagliata attraverso un lato con omega nullo. Se ne deduce che ogni cella, di quattro lati, può avere teoricamente al massimo quattro vicini e quindi otto celle interne, due prima riga interna e due seconda riga interna. La Figura 17 mostra un esempio di una cella tagliata, evidenziata in verde, e delle sue celle interne, colorate di

viola. Il pallino blu evidenzia una di queste celle interne mentre la sua riflessione é mostrata con un quadrato blu. Le celle marchiate con una x sono le tre celle i cui centri cella sono più vicini alla riflessione.



In output il preprocessore scrive quindi gli indici delle tre celle più vicine per le due celle interne per ognuna delle interfacce che da luogo a celle interne. Per evitare di allocare spazio inutile viene anche scritto quante sono queste celle interne ed è costruito un puntatore che indica, all'interno della matrice costituita da tutti gli indici delle celle interne, dove si trova l'inizio delle informazioni relativa ad ogni cella. Riassumendo quindi le informazioni addizionali previste nel metodo di Forrer sono: una matrice di sei coppie d'interi per il numero delle interfacce con omega nullo, ovvero gli indici i e j delle tre celle vicine ai centri cella della prima e delle tre celle vicine ai centri cella della seconda cella interna riflessi; una matrice di sei reali per il numero delle interfacce con omega nullo, che fornisce le distanze delle tre celle vicine alla riflessione della prima e delle tre celle vicine al posizione, all'interno delle due precedenti matrici, da cui partono le informazioni concernenti la cella d'indice dato.

Da notare che rispetto al precedente metodo non sono più passate le aree delle superfici non immerse delle celle ma è passato un termine denominato γ , che rappresenta la frazione dell'area immersa rispetto all'area totale della cella. I casi limite sono quindi di gamma pari all'unità per le celle esterne e gamma nullo per le celle interne.

Solutore

Il metodo di Forrer sfrutta il fatto che per un flusso non viscoso una boundary di parete si comporta come una linea di simmetria. Con riferimento alla Figura 18, se B é un punto della boundary e n é la sua normale orientata verso l'esterno, un punto P, appartenente allo spazio fisico, può essere riflesso in un punto r_p nella zona interna al di là della boundary attraverso la formula:

 $r_P = P - 2n(P - B)^T n$



Figura 18 Riflessione di un punto e del campo di variabili.

Per riflettere il valore di una variabile nel punto P si deve introdurre la matrice di riflessione R_{α} definita come:

$$R_{\alpha} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\alpha) & \sin(2\alpha) & 0 \\ 0 & \sin(2\alpha) & -\cos(2\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Il valore di una variabile nel punto $r_{\rm p}$ é dato da:

$Q(r_P) = R_{\alpha}Q(P)$

dove Q sono le variabili conservative, viste nel capitolo 2, ovvero (ρ , ρu , ρv , ρe). Calcolando in questo modo le variabili nelle celle interne, fantasma, la soluzione rispetta le equazioni e la condizione al contorno di parete.

Il metodo divide le celle in esterne, interne e tagliate. Le celle che sono attraversate dalla curva sono dette tagliate o di boundary. Dentro ogni cella la curva é approssimata, come nel metodo del capitolo precedente, da una retta. In Figura 19 é raffigurata una cella tagliata di indici (i,j). In giallo sono mostrate le due celle interne, che si ricorda sono celle per cui l'interfaccia della cella tagliata ha omega nullo. Il metodo ha bisogno della riflessione di queste due celle che in figura é rappresentato in colore viola. Per il calcolo dei flussi con uno schema del secondo ordine



Figura 19 Cella tagliata, celle interne e loro riflessione.

si ha uno stencil di quattro, il che significa che é necessario avere il valore delle variabili conservative in due celle dopo e due celle prima dell'interfaccia e quindi in due celle interne oltre al valore in tutte le celle esterne ed in quelle tagliate. Per il calcolo delle variabili nelle celle tagliate Forrer ha proposto di utilizzare una media pesata tra il valore nella parte denominata C_{ij}^2 (verde in figura) e il valore opportunamente riflesso della parte C_{ij}^1 (azzurra in figura). In formula:

$$Q_{ij} = \frac{1}{Vol_{ij}} \left[Q_{ij} Vol_{ij} \gamma_{ij} + R_{\alpha_{ij}} Q_{ij} Vol_{ij} (1 - \gamma_{ij}) \right]$$

in cui si ricorda che gamma é la frazione d'area non immersa rispetto all'area totale. Per la riflessione della variabile si utilizza la matrice R vista in precedenza.

Sempre riferendosi alla Figura 19, per determinare il flusso attraverso l'interfaccia i+1 oltre al valore delle variabili conservative in (i,j) ed in (i-1) é necessario avere anche il valore in (i+1,j) e (i+2,j), ovvero nelle due celle interne. Si prenda ora in considerazione la cella (i+1,j). La cella viene riflessa e s'individuano le tre celle i cui centri cella sono più vicini al centro cella della riflessa. Si ottengono a questo punto le variabili conservative nel centro cella riflesso attraverso un'interpolazione del valore delle variabili nelle tre celle più vicine. Una volta ottenute le variabili nel centro cella riflesso si opera la riflessione delle variabili, vista in precedenza, per ottenere i valori nel centro cella della cella interna. Volendo tradurre in formule quanto detto:

$$Q_{i+1,j} = R_{\alpha_{i,j}} Q_{\text{int } erp 3vicine}$$

in cui si utilizza un'interpolazione lineare per trovare le variabili nel centro cella riflesso:

$$Q_{\text{int erp3vicine}} = \frac{\sum_{k=1,3}^{k=1,3} dist_k - dist_1}{\sum_{k=1,3}^{k=1,3} dist_k} Q_1 + \frac{\sum_{k=1,3}^{k=1,3} dist_k - dist_2}{\sum_{k=1,3}^{k=1,3} dist_k} Q_2 + \frac{\sum_{k=1,3}^{k=1,3} dist_k - dist_3}{\sum_{k=1,3}^{k=1,3} dist_k} Q_3$$

in cui $dist_k$ sono le distanze tra il centro cella riflesso e il centro cella della cella k-esima. Per la cella (i+2) si procede nello stesso modo, partendo sempre dalla riflessione della cella verso l'esterno.

Risultati

Il metodo é stato completamente sviluppato ma si é ancora in fase di debugging. Si é, infatti, preferito dare priorità al metodo di Dadone che sembra essere più adatto ad essere applicato alle vele.

Capitolo 5 Il Metodo di Dadone e Grossman

Introduzione

É un metodo che, rispetto ai metodi trattati nei precedenti capitoli, ha la peculiarità di non avere celle tagliate, che oltre ad essere una complicazione in fase di preprocessing costituiscono un elemento d'instabilità in fase d'applicazione del metodo. Le forzanti che simulano la boundary sono infatti applicate nelle prime due celle, fantasma, che si trovano vicino alla parete di boundary. Il valore nei centri cella é assegnato mantenendo la condizione d'impermeabilità della boundary e soddisfacendo al tempo stesso l'equazione di conservazione della quantità di moto nella direzione normale alla boundary stessa.

Il metodo inoltre sembra essere abbastanza facilmente estendibile alle vele, ovvero a superfici (linee in 2d) di spessore nullo.

Preprocessore

Nel caso bidimensionale l'input del preprocessore é costituito da una griglia di background, che é la griglia su cui il solutore andrà a risolvere le equazioni della fluidodinamica, e da un certo numero di curve ognuna con il proprio attributo di curva chiusa o aperta. Un esempio di curva chiusa é la sezione di un albero mentre aperta é una sezione di randa o di fiocco. Per ogni curva é inoltre necessario dare anche una direzione di sweep lungo la griglia che quindi può essere solo una delle seguenti quattro:+i, -i, +j o –j.

In output il preprocessore fornisce per ogni curva due liste di celle interne prima riga, due liste di celle interne seconda riga e una serie di variabili associati a queste celle. In particolare i versori normali alla boundary, in quali quattro celle ricade l'immagine del centro cella riflessa rispetto alla boundary, i pesi delle quattro celle che comprendono l'immagine riflessa e le interfacce lungo la prima direzione di griglia (i) e lungo la seconda direzione (j) che sono influenzate nel solutore dalla presenza delle celle prima e seconda riga. In aggiunta, per le curve chiuse, è data una lista delle celle interne alla curva, la cui soluzione non ha significato fisico e in cui quindi la soluzione delle equazioni della fluidodinamica é superflua. Nel seguito saranno chiariti il significato di ognuna delle variabili che sono richieste in input o scritte in output o e a tal fine occorre comprendere più da vicino cosa fa il preprocessore. Per farlo ci si



Figura 20 Sezioni di base dell'albero e della randa di un catamarano classe Tornado

avvarrà di due esempi, uno per la curva chiusa e uno per quella aperta. In particolare si

utilizzeranno le sezioni di base i un albero e di una randa di un catamarano classe Tornado. In Figura 20 sono rappresentate queste sezioni. I quadrati sono i punti che definiscono le curve. Da notare che la scala é molto differente; la lunghezza lungo l'ascissa dell'albero é sui 14cm contro i 220 della randa.

Funzionamento del Preprocessore

In Figura 21 é schematizzato in un diagramma di flusso il funzionamento del preprocessore attraverso le sue routine principali. La prima operazione é la lettura della griglia di background. Questo implica la lettura delle dimensioni e l'allocazione dinamica della memoria necessaria ad ospitarne le coordinate. Il vantaggio di questo metodo rispetto agli usuali metodi bodyfitted é proprio nella facilità di generazione della griglia. Un esempio di griglia di background per l'albero e la randa sono in Figura 22. Il successivo input richiesto é il numero delle curve che si vogliono



processare all'interno del blocco di griglia di background scelto. Le successive routine sono applicate ad ognuna di queste curve. Il primo passo all'interno del ciclo sulle curve é la lettura della curva data come set di punti. Il







Figura 23 L'albero visto secondo due opposte direzioni di sweep.



successivo passo é il controllo delle dimensioni relative della curva rispetto alle dimensioni della griglia di background. Se la curva ha troppi pochi punti si provvede ad aumentarli. Il metodo per aumentarli é far passare una curva di tipo B-Spline attraverso i punti dati e valutarla in un numero di punti superiore a quello iniziale. Il risultato non perde quindi informazioni geometriche rispetto all'input iniziale. Nell'input della curva é presente anche la direzione di sweep, citata in precedenza. Ogni curva, sia aperta che chiusa, é sdoppiata nelle due facce, o side, in cui la si vede dalla direzione di sweep e dalla sua opposta. Nel caso di una curva chiusa le due side differiscono mentre nel caso della curva aperta coincidono. L'esempio di Figura 23 mostra come cambia una curva chiusa: in alto é raffigurata la curva totale all'interno della griglia di background, in mezzo la prima side ovvero la curva "vista" attraverso una direzione di sweep di -J (supponendo J l'indice nella direzione delle ordinate crescenti) e in basso la seconda side, curva "vista" dalla direzione opposta alla precedente i.e. +J. L'esempio per una curva aperta (la

randa) non é stato inserito perché sarebbe costituito banalmente da tre volte la stessa curva.

Una volta letta la curva e confrontata con la griglia di background vi é la routine che trova le celle della griglia di background che sono tagliate dalla curva. In pratica si cerca a quale cella appartiene ogni punto della curva. Una volta fatta la ricerca per tutti i punti si controlla che la sequenza di celle non presenti vuoti. In caso vi siano, si rimedia mettendo punti con valutazioni della B-Spline e colmando così i vuoti. La fase successiva é assegnare ad ogni cella tagliata una normale alla



boundary. Per fare ciò si utilizza l'associazione punti della curva-celle tagliate appena costruita e si procede ad una valutazione delle normale alla B-Spline nel punto dato. Quindi vi sono serie di routine una raggruppate sotto il nome blocco A. Queste si applicano ad entrambe le side. Per la side due vi sono altre due routine preliminari che semplicemente invertono la direzione di sweep



Figura 24 In alto ripartizione delle celle per l'albero. In basso ripartizione delle celle per la randa con zoom nel bordo d'attacco e nel bordo di uscita.

e nel caso della curva aperta, anche le normali. Nel caso della curva chiusa, infatti, le normali sono correttamente orientate essendo la side due, come si é visto in precedenza, solo una porzione della curva totale. Il blocco A é il cuore del programma per la comprensione dell'output. Le celle sono ripartite attraverso la direzione di sweep in interne, di boundary, esterne e speciali, si veda Figura 24 per i due casi di curva chiusa e aperta. Le celle di boundary trovate in precedenza che sono raggiungibili partendo dai confini del dominio e seguendo la direzione di sweep sono le nuove celle di boundary. Le celle che si attraversano per arrivare ad esse sono le celle esterne. Le celle non



Figura 25 Ripartizione delle celle per la prima e la seconda riga.

raggiunte mai sono celle interne. Le celle attraversate per arrivare da un confine all'altro del dominio, sempre seguendo la direzione di sweep, sono le celle speciali.

La ripartizione vista é quella valida per trovare le celle della seconda riga interna. Per le celle della prima riga interna la ripartizione é quella in Figura 25. Come si può notare la differenza é solo ai punti estremi della curva, lungo le ascisse per uno sweep lungo le ordinate.

Per la curva aperta valgono le stesse considerazioni e la figura non é riproposta. La ripartizione serve a trovare le celle interne su cui il solutore applica un trattamento speciale. Le celle della prima riga (interna) sono le prime celle il cui centro cella giace dalla parte interna della boundary. Questo si traduce nelle celle di boundary, verdi in Figura 26, se il centro cella ricade verso l'interno, oppure in una o più delle celle interne che si raggiungono a croce, ovvero spostandosi lungo le due direzioni i e j. Ecco perché la ripartizione delle celle per la prima riga prevede quelle due strisce blu che prevengono dal trovare delle false celle (prima riga). Ad ogni cella prima riga è associata la normale della cella tagliata da cui proviene. La ricerca delle celle per la seconda riga interna si avvale delle celle della prima riga; vengono prese come celle della seconda riga tutte le celle che andando a croce partendo dalle celle della prima riga sono interne. Ad ogni cella della seconda riga è associata una normale media delle normali della prima cella da cui la cella proviene. Nella Figura 26 e Figura 27 sono mostrate le celle prima e seconda riga per i due esempi di curva aperta e chiusa. Nel caso delle celle della prima riga in blu sono le celle di boundary con il centro cella interno mentre in rosso sono le celle interne prese come prima riga.



Figura 26 Celle della prima riga per l'albero, in alto, e per la randa, in basso.





Figura 27 Celle della seconda riga per l'albero, in alto, e per la randa, in basso.

Una volta trovate le celle della prima della e seconda riga interna né si fa il mirror, ovvero la riflessione, dei centri cella rispetto alla boundary, ovvero si i trovano punti ribaltandoli come in uno specchio rispetto alla curva. Per ognuno di questi punti ribaltati si trovano le quattro celle che lo contengono e si costruiscono i pesi per l'interpolazione bilineare.

Nel solutore il valore della variabile nel punto ribaltato sarà ottenuto come media pesata dei valori delle quattro celle che lo contengono. I pesi sono costituiti dalle distanze dei centri cella delle quattro celle dal punto ribaltato.

In Figura 28 sono rappresentati i centri delle celle







della prima e seconda riga con relativi mirror per i due esempi.

Il successivo passo, sempre all'interno del blocco di subroutine A, é la determinazione delle interfacce i cui flussi sono condizionati dai valori delle variabili dalle celle della prima e seconda riga. Le interfacce sono divise in interfacce i e interfacce j. Nel solutore queste interfacce dovranno



Figura 29 Sopra: Interfacce I, a sinistra, e interfacce J, a destra, per l'albero. Sotto e pagina successiva: interfacce I e interfacce J per la randa.

essere trattate a parte e quindi nasce l'esigenza di isolarle. Il solutore utilizza uno schema symmmetric tvd o upwind tvd del secondo ordine e richiede uno stencil di quattro, il valore delle variabili in due celle a sinistra(sotto per le interfacce j) dell'interfaccia considerata ed in due a destra (sopra per le interfacce j),. In Figura 29 sono raffigurate in rosso le interfacce trovate. In realtà sono illuminate le celle la cui interfaccia a destra(sotto) é una di quelle trovate.







Le subroutine viste in precedenza sono applicate anche per la side due. Per brevità sono proposte solo le figure riguardanti l'albero.



Figura 30 Ripartizione delle celle per la side due dell'albero.



Figura 31 Celle della prima e della seconda riga per la side due dell'albero.



Figura 32 Prima e seconda riga e relativi mirror per la side due dell'albero.



Figura 33 Interfacce I, a sinistra, e interfacce J, a destra, per la side due dell'albero.

Infine nel caso di curve chiuse vi é la determinazione delle celle interne alla curva totale, o se si preferisce alla somma della curva vista dalla direzione di sweep con la curva vista dalla direzione



opposta. Le celle interne saranno l'insieme intersezione delle celle interne alle due side. In Figura 34 sono mostrate le celle interne all'albero.

Solutore

Il solutore come primo passo deve allocare le variabili necessarie per il metodo e leggere tutti i dati che il preprocessore gli fornisce via file. In Figura 35 vi é un riassunto dei dati provenienti dal preprocessore.





Le variabili di figura sono state tutte descritte nel preprocessore. In aggiunta a queste nel solutore si allocano tre vettori, con dimensione numero di celle, chiamati multinti, multintj e multtot ed una matrice contenente le quattro variabili primitive (ρ ,u,v,p) per tutte le celle prima e seconda riga. I primi due vettori sono riempiti con il risultato di un "and" logico tra i vari moltiplicatori per interfacce. In altre parole, ad esempio, multinti assume il valore zero laddove a quel determinato (i,j) corrisponda una interfaccia (cioé intii=i, intij=j) in una qualsiasi curva e side, e il valore uno in tutte le altre celle. Il terzo vettore é riempito con un "and" logico dei moltiplicatori (mult) che

definiscono le celle interne quindi nel caso tipico di un albero, una randa ed un fiocco coincide con il moltiplicatore dell'albero. Per la matrice con le quattro variabili primitive in seguito si descriverà in dettaglio come viene riempito.

Una volta letto e inizializzato tutte le variabili menzionate l'unica parte modificata rispetto al solutore originale é nel calcolo dei flussi per il blocco contenente le curve immerse. In Figura 36 sono mostrate le routine coinvolte nel calcolo dei flussi convettivi per uno schema del secondo ordine simmetrico. Il diagramma per gli altri schemi é equivalente.

Routine

IBMSYMTVD



Figura 36 Routine coinvolte nel calcolo dei flussi convettivi per un blocco con curve immerse.

La prima routine che è eseguita, ovvero IBMSYMTVD, esegue il calcolo secondo lo schema dei flussi convettivi per tutte le interfacce non influenzate dalle celle interne della prima o seconda riga . In pratica si avvale dei tre vettori globali menzionati prima (multinti, multintj, multtot) per annullare i contributi provenienti da interfacce influenzate dalle celle delle prime due righe interne o i contributi alle celle che sono interne alla curva completa.

Il metodo quindi alloca una matrice temporanea per le quattro variabili primitive $(\rho_{temp}, u_{temp}, v_{temp}, p_{temp})$ che sarà utilizzato nelle successive routine dentro ai due loop o cicli e poi scartato. Il passo successivo sono i due loop il primo sulle curve ed il secondo più interno sulle side. All'interno è chiamata per prima la routine IBMFIELDTEMP, che si occupa di riempire la matrice temporanea con, nel seguente ordine, le variabili primitive globali e le variabili primitive associate alle celle della prima e seconda riga. La routine successiva, chiamata IBMBC, é il cuore del metodo. La routine per ogni cella della prima e seconda riga fa le seguenti cose: calcola il valore delle variabili conservative nel punto riflessione del centro cella della cella considerata, attraverso un'interpolazione bilineare sui valori nelle quattro celle contigue più vicine al punto riflesso;

determina il nuovo valore delle variabili primitive nella cella; salva il risultato nella matrice delle variabili della cella e nella matrice globale temporanea. L'interpolazione bilineare non merita attenzione ma invece occorre soffermarsi sulla determinazione del valore delle variabili primitive nella cella partendo dalle variabili conservative nel punto riflesso. Quello che interessa non é la trasformazione da variabili conservative a primitive e viceversa, che nel caso di gas perfetti é banale e che in ogni modo non contiene informazioni del metodo, ma la trasformazione dal punto riflesso al centro cella originale. Per fare ciò il metodo utilizza quelle che vengono chiamate condizioni al contorno CCST e, che in formula, sono:

$$\rho_{i} = \rho_{r} \left(\frac{p_{i}}{p_{r}}\right)^{1/\gamma}$$

$$V_{\tan i}^{2} = V_{\tan r}^{2} + \frac{2\gamma}{\gamma - 1} \left(\frac{p_{r}}{\rho_{r}} - \frac{p_{i}}{\rho_{i}}\right) + V_{norm_{r}}^{2} - V_{norm_{i}}^{2} \quad (3)$$

$$V_{norm_{i}} = -V_{norm_{r}}$$

$$p_{i} = p_{r} - \rho_{r} \frac{V_{\tan r}^{2}}{R_{r}} \Delta n_{i \rightarrow r}$$

in cui i pedici i e r indicano rispettivamente la cella interna e la cella riflessa nel campo esterno; V_{tan}, V_{norm} sono le velocità tangente e normale alla boundary; R_r é il raggio di curvatura della boundary nel punto intersezione tra la retta congiungente il centro cella ed il suo riflesso e la bondary; $\Delta n_{i->r}$ é la distanza tra il centro cella e la sua riflessione. Nell'implementazione fatta sì é preferito considerare un raggio di curvatura locale infinito, parete piana, il che semplifica molto le relazioni (3) che diventano:

$$\rho_{i} = \rho_{r}$$

$$V_{\tan i} = V_{\tan r}$$

$$V_{normi} = -V_{normr}$$

$$p_{i} = p_{r}$$

A questo punto sono note per la data curva e side tutte le variabili nelle celle interne ed esterne e si può procedere al calcolo dei flussi con lo schema scelto (in questo esempio il simmetrico) nelle interfacce influenzate dalle celle della prima, seconda riga e che erano state ignorate dalla prima routine esterna ai loop. Per comodità questa parte é stata divisa in due routine, IBMFLUXISYM e IBMFLUXJSYM, che calcolano i flussi rispettivamente per le interfacce lungo i e le interfacce lungo j.

Risultati

Come nei precedenti metodi si é scelto come caso test il cilindro in basso subsonico. In Figura 37 sono riportati i confronti tra il campo di pressione calcolato con un metodo bodyfitted ed uno con il presente metodo e si riferiscono rispettivamente ad uno schema simmetrico del secondo ordine ed uno schema upwind del secondo ordine.



Figura 37 A sinistra confronto tra metodo bodyfitted (alto) e metodo di Dadone (basso) con schema simmetrico del secondo ordine. A destra confronto tra metodo bodyfitted (alto) e metodo di Dadone (basso) con schema upwind del secondo ordine.

Si può notare che se per lo schema simmetrico il metodo di Dadone sembra fallire nella determinazione del recupero di pressione al bordo d'uscita (a 180 gradi), per lo schema simmetrico il metodo di Dadone sembra fornire i migliori risultati. Il metodo é stato provato con successo anche sulla sezione dell'albero di un catamarano classe Tornado (Figura 38) mentre si é in fase di debugging per l'applicazione a curve aperte, come la randa ad esempio.



Figura 38 Confronti tra bodyfitted (destra) e metodo di Dadone(sinistra) per la sezione dell'albero di un catamarano classe Tornado. Dall'alto in basso schemi del secondo ordine upwind, secondo ordine simmetrico, primo ordine.

Bibliografia

- 1. Vieceli J.A., "A method for including arbitrary external boundaries in the MAC incompressible fluid computing technique", J.. Comput. Phys., 4, 543-551, 1969
- 2. Welch J.E., Harlow F.H., Shannon J.P. and Daly B.J., "A computing technique for solving viscous incompressible transient fluid flow problems involving free-surfaces", Report LA-3425, Los-Alamos Scientific Lab., 1966.
- 3. Harlow F.H. and Welch J.E., "Numerical calculation of time dependent viscous incompressible flows of fluid with free surface", Phys. Fluids, **8**, 2182, 1965
- Vieceli J.A., "A computing method for incompressible flows bounded by moving walls", J. Comput. Phys., 8, 119-143, 1971
- Peskin C.S., "Flow patterns around heart valves: a digital computer method for solving the equation of motion", Ph.D. thesis, Physiology, Albert Einstein College of Medicine, Univ. Microfilms 72-30, 378, 1977
- Peskin C. S., "Numerical analisys of the blood flow in the heart", J. Comput. Phys., 25, 220-252, 1977
- Peskin C. S., McQueen D.M., "A three-dimensional computational method for blood flow in the heart: (I) immersed elastic fibers in a viscous incompressible fluid", J. Comput. Phys., 81, 372-405, 1989
- McQueen D.M., Peskin C. S., "A three-dimensional computational method for blood flow in the heart: (II) contractile fibers", J. Comput. Phys., 82, 289-297, 1989
- Basdevant C., Sadourny R., "Numerical solution of incompressible flow: the mask method", Lab. De Metereologie Dynamique, Ecole Normale Superieure, Paris (unpublished)
- 10. Briscolini M., Santangelo P., "Development of the mask method for incompressible unsteady flows", J. Comput. Phys., 84, 57-75, 1989
- Goldstein D., Handler R., Sirovich L., "Modeling no-slip flow boundary with an external force field", J. Comput. Phys., 105, 354-366, 1993
- 12. Saiki E.M., Biringen M., "Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method", J. Comput. Phys., **123**, 450, 1996
- LeVeque R.J., Clahoum D., "Cartesian grid methods for fluid flow in complex geometries", Computational Modeling in Biological Fluid Dynamics, L.J. Fauci and S. Gueron (eds) IMA Volumes in Mathematics and its Applications, **124**, 117-143, 1999
- 14. Lai M.C., Peskin C.S., "An immersed boundary method with formal second order accuracy and reduced numerical viscosity", J. Comput. Phys., **160**, 705-719, 2000

- 15. Mohd-Yosuf J., "Combined immersed boundary/B-Spline methods for simulation of flow in complex geomteries", Annual research Briefs, Center for Turbulence Research, 317-328, 1997
- 16. Peskin C. S., "The immersed boundary method", Acta Numerica, 479-517, 2002
- Iaccarino G., Verzicco R., "Immersed boundary technique for turbulent flow simulations", App. Mech. Rev., vol. 56, no 3, May 2003
- 18. Kalizin G., Iaccarino G., "*Toward immersed boundary simulation of high Reynolds number flows*", Center for Turbulence Research Annual Research Briefs 2003
- 19. Ming-Chih-Lai, Peskin C.S., "An immersed boundary method with formal second-order accuracy and reduced numerical viscosity", J. Comput. Phys., 160, 705-719, 2000
- Fadlum E.A., Verzicco R., Orlandi P., Mohd-Yusof J., "Combined immersed boundary finitedifference methods for three-dimensional complex flow simulations", J. Comput. Phys., 163, 35-60, 2000
- 21. Khadra K., Parneix S., Angot P., Caltagirone J.P., "*Fictious domain approach for numerical modeling of Navier-Stokes equations*", 4th Int. Conf. On Navier-Stokes Equations and Related Nonlinear Problem, 1995
- 22. Angot P., Bruneau C.H., Frabrie P., "A penalization method to take into account obstacles in viscous flows", Numerische Mathematik, **81**, 497-520, 1999
- 23. Kevlahan N., Ghidaglia J.M., "Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization", Eur. J. Mech. B/Fluids, **20**, 333-350, 2001
- 24. Berger M., Aftosmis M., "Aspects (and aspect ratios) of Cartesian mesh methods", 16th Int. Conf. On Numerical Methods in Fluid Dynamics, 1998
- Ye T., Mittal R., Udaykumar H.S., Shyy W., "An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries", J. Comput. Phys., 156, 209-240, 1999
- 26. Coirier W.J., "An adaptively-refined, Cartesian, cell-based scheme for the Euler and Navier-Stokes equations:, NASA TM106754, 1994
- Almgren A.S., Bell J.B., Colella P., Marthaler T., "A Cartesian grid projection method for the incompressible Euler equations in complex geometries", SIAM J. Sci. Comput. (USA), 18, 1289-1309
- 28. Hirt C.W., Nichols B.D., "Volume of fluid (VOF) method for the dynamics of free-boundaries", J. of Comp. Phys., 39, 201-225, 1981
- 29. Sethian J.A., "Level-set methods and fast marching methods", Cambridge Univ. Press, 1996
- 30. Kim J., Kim D., Choi H., "An Immersed Boundary Finite-Volume Method for Simulation of Flows in Complex Geometries", J. Comput. Phys., **171**, 132-150, 2001

- Verzicco R., Fatica M., Iaccarino G., Moin P., Khalighi B., "Large Eddy Simulation of a Road-Vehicle with Drag Reduction Devices", Center for Turbulence Research Annual Research Briefs 2000
- 32. Verzicco R., Mohd-Yusof J., Orlandi P., Haworth D., "Large Eddy Simulation in Complex Geometric Configurations Using Boundary Body Forces", AIAA J., vol. 38, no 3, March 2000
- 33. Pan K.L., Shy W., Law C.K., "An Immersed-boundary method for the dynamics of premixed flames". Int. J. of Heat and Mass Transfer, **45**, 2002, 3503-3516
- 34. Mohd-Yusof J., "*Development of Immersed Boundary methods for Complex Geometries*", Center for Turbulence Research Annual Research Briefs 1998
- 35. Iaccarino G., Kalitzin G., Khalighi B., "Towards an Immersed Boundary Flow Solver", AIAA 2003-0770, 41st Aerospace Sciences Meeting & Exhibit, 2003
- 36. Tessicini F., Iaccarino G., Fatica M., Wang M., Verzicco R., "Wall Modeling for Large-Eddy Simulation using an Immersed Boundary Method", Center for Turbulence Research Annual Research Briefs 2002
- 37. Mohd-Yusof J., "Combined Immersed Boundary/B-spline Methods for Simulation of Flow in Complex Geometries", Center for Turbulence Research Annual Research Briefs 1997
- 38. Kalitzin G., Iaccarino G., "*Turbulence Modeling in an Immersed-Boundary RANS Method*", Center for Turbulence Research Annual Research Briefs 2002
- 39. Majumdar S., Iaccarino G., Durbin P., "*RANS Solvers with Adaptive Structured Boundary Non-Conforming Grids*", Center for Turbulence Research Annual Research Briefs 2001
- 40. Verzicco R., Mohd-Yusof J., Orlandi P., Haworth D., "LES in Complex Geometric Configurations Using Boundary Body Forces", Center for Turbulence Research, Proceedings of the Summer Program 1998
- 41. Verzicco R., Iaccarino G., Fatica M., Orlandi P., "Flow in an Impeller Stirred Tank Using an Immersed Boundary Method", Center for Turbulence Research Annual Research Briefs 2000
- 42. Richards K.J., Pomraning E., Senecal P.K., Choi C.Y., Rutland C.J., "On the Use of the Immersed Boundary Method for Engine Modeling"
- 43. Kim Y., Peskin C., "2-D Parachute Simulation by the Immersed Boundary Method"
- 44. Kim Y., Peskin C., "Penalty Immersed Boundary Method for an Elastic Boundary with Mass"
- 45. Cortez R., Minion M., "The Blob Projection Method for Immersed Boundary Problems", Int. J., Comput. Phys., **161**, 428-453, 2000
- 46. Tyagi M., Acharya S., "Large Eddy Simulation of Complex Turbulent Flows Using Immersed Boundary Method"

- 47. Tsukanov I., Shapiro V., Zhano S., "A Meshfree Method for Incompressible Fluid Dynamics Problems", Sail 2002, 1, Spatial Automation Laboratory
- 48. Dadone A., Grossman B., "Ghost-Cell Method for Inviscid Two-Dimensional Flows on Cartesian Grids", AIAA J., 42, 12, December 2004
- 49. Forrer H., Jeltsch R., "A Higher-Order Boundary Treatment for Cartesian-Grid Methods", J. Comput. Phys., **140**, 259-277, 1998
- 50. Forrer H., "Boundary Treatment for a Cartesian Grid Method", Research Report no 96-04, April 1996, Seminar für Angewandte Mathematik Eidgenössische Technische Hochschule CH-8092 Zürich Switzerland
- 51. Batina J.T., "A gridless Euler/Navier-Stokes algorithm for complex aircraft applications", AIAA Paper 93-0333, Jan. 1993
- Subrata R., Fleming M., "Nonlinear subgrid embedded element-free Galerkin method for monothone CFD solutions", Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conf., FEDSM99-7023, Jul. 1999
- 53. Belytschko T., Lu Y.Y., Gu L., "*Element-free Galerkin method* ", Int. J. of Numerical Methods in Engineering, **37**, 2, 229-256, 1994
- 54. Koh E.P.C., Tsai H.M., Liu F., "Euler solution using Cartesian grid with a gridless least-squares boundary treatment", AIAA J., 43, 2, 246-255
- 55. Kirshman D.J., Liu F., "A gridless boundary conditions method for the solution of the Euler equations on embedded Cartesian meshes with multigrid", J. of Comput. Phys. 201, 1, 119-147, 2004