

INTERAZIONE FLUIDO-STRUTTURA: GENERAZIONE DI GRIGLIE DI CALCOLO *

Marco CAPOZZI **

* *The Author wishes to thank Dr. William M. Chan together with the staff from NASA Ames Research Center and Dr. Anders Petersson, from CASC –Center for Applied Scientific Computing, who have allowed the use of some of the pictures appearing in this article.*

L'articolo presenta alcune nozioni sulla generazione di griglie di calcolo, dette anche mesh o grid. Data la notevole complessità dell'argomento vengono descritti i concetti di base della materia, analizzando le tecniche e le difficoltà più comuni. (si rimanda testi specifici l'eventuale approfondimento da parte del lettore). Nonostante l'articolo sia stato scritto con riferimento alle applicazioni della CFD, i contenuti descritti sono comunque validi anche in altri ambiti delle simulazioni numeriche, come ad esempio quello strutturale o meccanico. L'Autore desidera ringraziare vivamente il Dr. Anders Petersson (attualmente presso il CASC- Center for Applied Scientific Computing), per aver concesso l'uso delle immagini dei grid da lui generati, e il Dr. William M. Chan del NASA Ames Research Center per aver autorizzato l'uso delle immagini dei grid Chimera.

** *Ingegnere Meccanico; Master in "Science in Aerospace Engineering", Mississippi State University (USA)*

INTRODUZIONE

La soluzione delle equazioni differenziali alle derivate parziali (*Partial Differential Equations, PDE*) tramite metodi numerici necessita di *griglie di calcolo*, comunemente dette *grid* o *mesh*. La griglia di calcolo è una decomposizione dello spazio in cui si vuole studiare una PDE in domini elementari semplicemente connessi. Una retta orientata suddivisa in un numero N di segmenti è un grid, e tale tipo di decomposizione fu usata per decine di anni prima dell'invenzione dei computer. Ovviamente, tali grid venivano utilizzate per usi puramente accademici.

La teoria sulla generazione dei grid iniziò a svilupparsi intorno agli Anni Settanta quando videro la luce i primi metodi numerici per la risoluzione dei flussi potenziali per applicazioni aeronautiche. Tali metodi, detti metodi a pannelli, ricadevano nella categoria BEM (*Boundary Element Method*). Poiché i metodi BEM guardano solo la superficie fisica del corpo considerato, la necessità di grid è limitata ad una discretizzazione bidimensionale del contorno del corpo in questione anche nel caso di corpi tridimensionali. Infatti la risoluzione del problema avviene su un pannello, ossia un guscio avente spessore nullo.

Per esempio: un aereo può essere rappresentato come un guscio che approssima la sua superficie esterna. All'infittirsi del grid la differenza fra superficie reale ed approssimata tende ad annullarsi. La tridimensionalità di un problema viene restituita dalle normali alle superfici dei pannelli, che variando in direzione e verso causano un orientamento delle superfici tale da creare una superficie composta 'tridimensionale'.

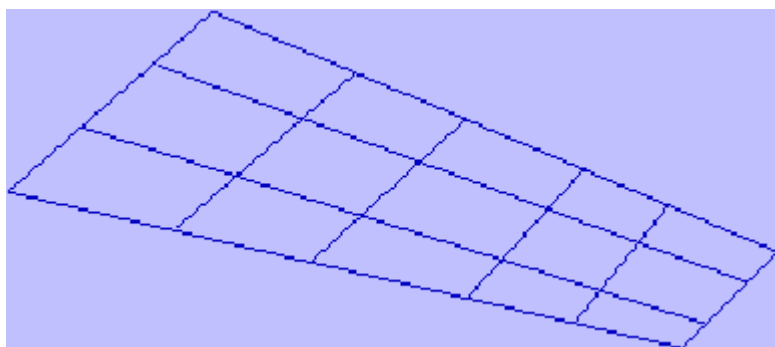


Figura 1

La Figura 1 riporta un esempio di pannellatura di un'ala di aereo effettuata per calcoli BEM: si osservi che pur essendo il problema tridimensionale, l'ala è bidimensionale (non necessariamente piatta!). Le normali ai singoli pannelli (i tasselli che compongono l'ala) servono a definire la superficie nel senso fisico, ossia: non può esserci flusso attraverso la superficie dell'ala, al più il flusso è tangente alla stessa. In questo modo l'ala è trattata nella sua tridimensionalità pur restando la griglia bidimensionale.

Con l'avvento dei moderni calcolatori digitali e l'aumento di memoria, velocità dei processori e loro parallelismo operativo, e con il contemporaneo progredire delle tecniche numeriche, si è approdati ai *metodi di campo*. Tali metodi, a differenza dei BEM, consentono la risoluzione del campo di moto di un fluido in una regione di spazio fisico che comprende il corpo, fornendo quindi una visione completa del campo di moto. È però fondamentale disporre di grid decisamente più complessi rispetto a quelli dei metodi a pannelli, essendo richiesta la capacità di descrivere geometrie bi o tri dimensionali nella loro interezza. Da qui (Anni Ottanta) si sviluppò la grid generation, ossia una branca della meccanica computazionale il cui solo scopo è quello di creare mezzi per discretizzare il dominio di definizione delle PDE. Se ne discute di seguito.

GENERAZIONE DI GRID

La generazione di grid è uno degli aspetti più delicati della fluidodinamica numerica: un cattivo grid, infatti, conduce sicuramente ad una soluzione quantomeno approssimata, se non del tutto errata. Un grid di buona qualità viceversa consente alle volte anche di accelerare la soluzione, oltre ad aumentare attendibilità della soluzione determinata.

I grid possono essere di due tipi: strutturati e non strutturati. I grid non strutturati sono relativamente recenti (primi anni novanta) e rappresentano oggi un ramo molto promettente della grid generation in funzione della semplicità da essi offerta nella generazione dei grid, cosa molto appetibile da un punto di vista commerciale, ma anche apprezzata da chi debba eseguire analisi numeriche.

I primi grid adottati furono quelli Cartesiani, che consistono in una suddivisione in celle quadrangolari (rettangoli o quadrati) del dominio di calcolo.

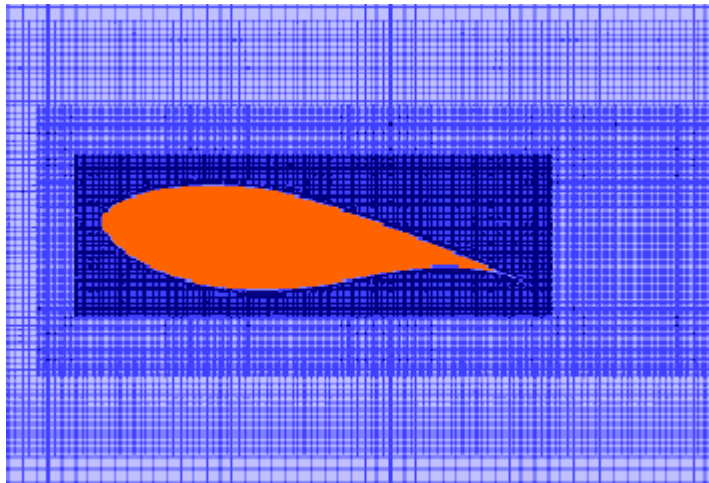


Figura 2

Ovviamente in tre dimensioni si fa uso di cubi o parallelepipedi. Nonostante questi grid siano i più semplici da generare, il loro uso in caso di superfici non planari costringe ad approssimare le superfici con elementi piccolissimi, oppure a ricorrere ad algoritmi interpolatori che di norma sono funzione della geometria da descrivere. Il principale vantaggio dei grid Cartesiani è dato dalla semplicità di programmazione.

La Figura 2 rappresenta un esempio di mesh Cartesiana generata sulla superficie di un profilo alare: si confronti la densità della griglia sul bordo del profilo rispetto a quella della Figura 3 (Grid Non Strutturato su Profilo NACA 0012). Attualmente i grid Cartesiani non sono molto usati.

I grid di tipo body conforming o body fitted, noti anche col termine di *grid strutturati* (*Structured Grids*), rappresentano l'evoluzione dei grid Cartesiani. In tale tipo di grid si definiscono delle coordinate curvilinee che seguono l'andamento del corpo (da cui la dicitura body fitted). Diversamente da quanto accade con i grid di tipo Cartesiano, in questo caso la PDE non può essere risolta nello spazio fisico, bensì è necessario ricorrere ad uno spazio computazionale. L'esistenza di due spazi costringe all'uso di trasformazioni di spazi vettoriali, ed impone anche una trasformazione delle condizioni al contorno. Esistono vari modi di generare un grid di tipo strutturato, più avanti si forniranno dei cenni in merito.

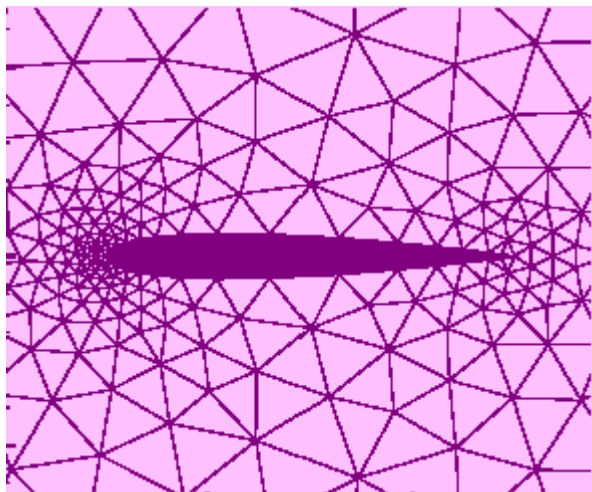


Figura 3

I grid di tipo non strutturato sono costituiti da triangoli in due dimensioni e da piramidi o tetraedri in tre dimensioni. Le griglie non strutturate offrono la possibilità di generare mesh in maniera quasi automatica con un intervento ridotto al minimo da parte dell'operatore: è questo un vantaggio non trascurabile che le altre tipologie di grid non hanno, poiché utenti anche non esperti di grid generation possono risolvere problemi su superfici complesse. Anche i grid di tipo non strutturato sono affetti da svantaggi che verranno analizzati meglio in seguito.

In applicazioni di tipo fluidodinamico – si vedrà di seguito – è spesso conveniente adottare grid ibridi, ovvero di tipo strutturato sulla parete del corpo e di tipo non strutturato nel resto del volume di controllo in cui si ricerca la soluzione delle PDE considerata.

Discretizzazione dei Domini di Calcolo

Prima di passare ad un'analisi più dettagliata delle varie tipologie di grid, è bene esporre alcuni concetti di base sulla generazione di griglie.

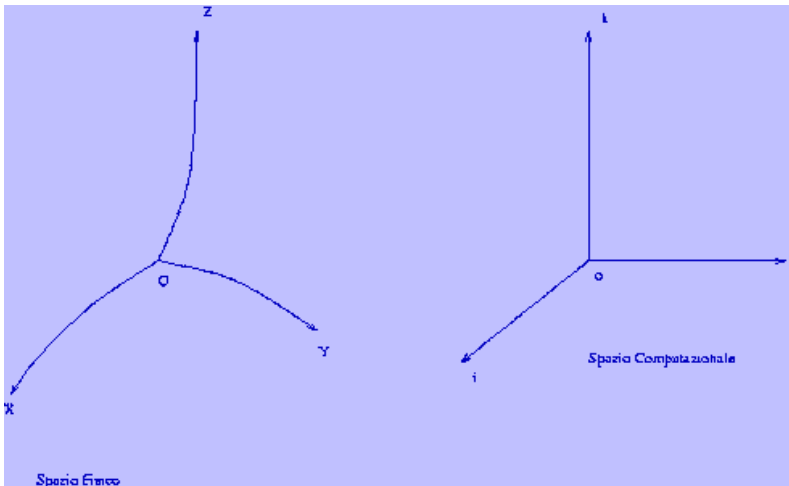


Figura 4

Per definizione, una griglia è una suddivisione di uno *spazio fisico* (Figura 4, immagine a sinistra). Lo spazio fisico è lo spazio in cui avvengono i fenomeni che si vogliono analizzare. Tipicamente tale spazio viene indicato con una terna curvilinea ortonormale (x_1, x_2, x_3) , che nel caso limite diventa un riferimento Cartesiano.

Lo *spazio computazionale* (Figura 4, immagine a destra) è quello invece in cui vengono calcolate le PDE. Una volta risolta la PDE nello spazio computazionale, la soluzione viene riportata nello spazio fisico mediante opportune trasformazioni.

La generazione del grid prevede l'esecuzione di due passi intermedi. Il primo consta nella creazione di uno *spazio di distribuzione*. Lo spazio di distribuzione è una sequenza di numeri creata tramite una funzione monodroma che ad ogni punto dello spazio fisico ne associa uno dello spazio di distribuzione. Dati:

$$(1) \quad (x_i, y_i, z_i) \quad i = 1, 2, \dots, n$$

e una sequenza di numeri:

$$(2) \quad r = (r_1, r_2, \dots, r_n), \\ 0 \leq r_j \leq 1, r_1 = 0, r_n = 1$$

se esiste una funzione che leghi univocamente i punti dello spazio definiti in (1) alle sequenze definite in (2), tale distribuzione prende il nome di *distribuzione curva di punti*. Ampliando il concetto a spazi a due o tre dimensioni si definiscono le *distribuzioni superficiali e volumiche di punti*. Nella generazione di grid si fa ricorso allo spazio di distribuzione dei punti per poi creare uno spazio computazionale di soli numeri naturali.

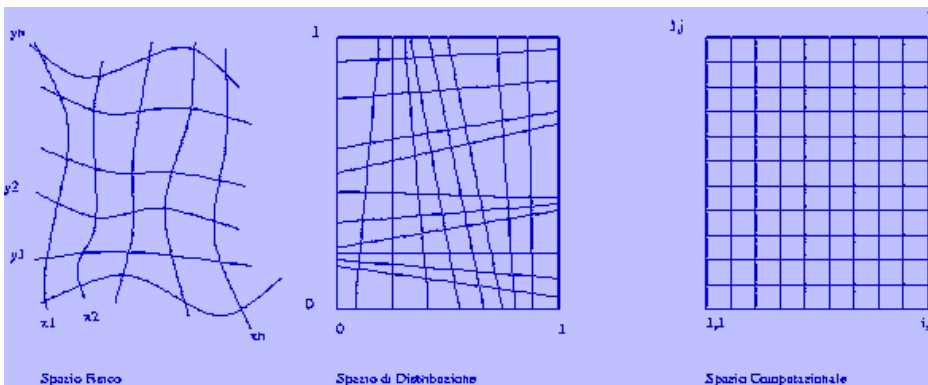


Figura 5

Si faccia riferimento, in proposito, alla Figura 5: notare come varino i valori e la distribuzione spaziale delle coordinate dei punti passando dallo spazio fisico a quello di distribuzione e da questo al computazionale. Ad ogni passaggio è associato uno *Jacobiano* da risolvere (ed invertire) per via numerica con tutti i problemi ad essa associati.

Il processo di generazione di reticoli di calcolo coinvolge molti aspetti: algoritmi, programmazione, capacità di calcolo, tipo di grid, ecc. Bharat Soni propone i seguenti punti chiave di cui tenere conto per analizzare criticamente la generazione dei grid:

1. *Facilità della generazione del grid;*
2. *Qualità del grid;*
3. *Trattamento delle condizioni al contorno;*
4. *Grandezza media delle celle che compongono il grid;*
5. *Facilità di discretizzazione della PDE;*
6. *Potenza dei computer;*
7. *Ore-uomo necessarie alla creazione del grid;*
8. *Complicazioni geometriche derivanti dalla geometria da descrivere;*

9. Facilità di uso del software;
10. Limiti dello schema di simulazione adottato;
11. Efficienza della simulazione effettuata;
12. Accuratezza della simulazione;
13. Possibilità di impiego di metodologie multigrid;
14. Adattabilità;
15. Possibilità di parallelizzazione del codice.

Il processo di generazione di una mesh, indipendentemente dalla tipologia, consta dei seguenti passi fondamentali:

a) **Mappatura della geometria:** viene stabilita una corrispondenza biunivoca fra lo spazio fisico e quello computazionale, utilizzando ove necessario suddivisioni multiblocco nel caso di griglie strutturate o di tipo ibrido. Nel caso di grid non strutturati o di tipo ibrido è anche necessario stabilire una relazione d'ordine fra i nodi delle celle.

b) **Generazione della geometria:** le superfici ed i volumi che definiscono la geometria da analizzare vengono descritte numericamente. Tale descrizione numerica comprende sia la mappatura che l'associatività dei punti.

c) **Modellazione computazionale:** viene generato un grid attorno alle superfici suddette. Nel caso di mesh strutturate vanno coniugate strategia multiblocco (funzione della geometria in questione), distribuzione di punti, regolarità della superficie da descrivere (smoothness in inglese, levigatezza in italiano, ma l'uso del termine ci sembra improprio) e ortogonalità delle celle. Per quanto concerne le mesh non strutturate occorre controllare sia la distribuzione dei punti sulla superficie che la 'crescita' della mesh (si chiarirà meglio in seguito il significato di tale parola).

Si consideri che il 90% del tempo consumato nella generazione di un reticolo di calcolo è speso sulla corretta distribuzione di punti sulle superfici da cui verrà generato il grid. Tale compito è indubbiamente noioso, ma di fondamentale (e sottovalutata) importanza per la qualità del grid generato e, di conseguenza, della soluzione della PDE.

GRIGLIE STRUTTURATE

Metodi Algebrici

La generazione di grid mediante metodi algebrici è ottenuta attraverso l'impiego di tecniche di interpolazione al fine di creare una distribuzione di punti nello spazio comprendente sia i punti sul contorno che quelli interni al dominio. Dato che le mesh vengono create a partire da relazioni funzionali più o meno semplici, i metodi algebrici sono in generale efficienti da un punto di vista computazionale.

I grid di tipo strutturato ricadono in due categorie: quelli Cartesiani e quelli a mappatura conforme (grid curvilinei). Considerati lo spazio e la geometria da descrivere, siano $\mathbf{r}()$ ed $\mathbf{W}()$ due riferimenti. È possibile 'saltare' da un riferimento ad un altro mediante una opportuna trasformazione. Lo spazio fisico è rappresentato dal vettore \mathbf{r} , quello computazionale dal vettore \mathbf{W} . Per mesh Cartesiane i due riferimenti coincidono e la trasformazione è una identità. In tal caso, infatti, spazio computazionale e geometrico sono la stessa cosa.

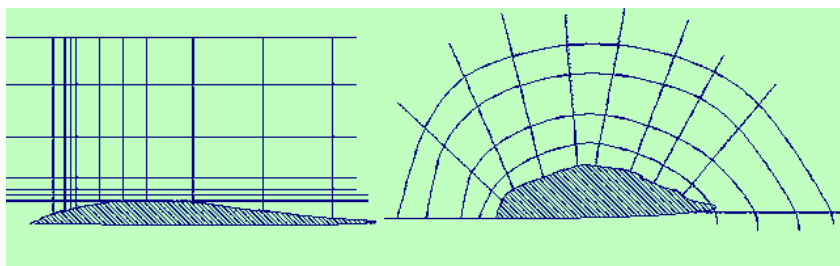


Figura 6

Le griglie Cartesiane offrono indubbiamente una grande facilità di programmazione opposta, però, ad una estrema difficoltà (se non impossibilità) nel descrivere superfici che non siano planari. Si veda la Figura 6 nella quale è riportato il confronto fra Grid Cartesiano (a sinistra) e Grid a Mappatura Conforme (a destra)

I grid di tipo curvilineo consentono di aggirare l'ostacolo rendendo necessario il ricorso a trasformazioni geometriche (endomorfismi). Tali trasformazioni coinvolgono il calcolo numerico di uno Jacobiano, e ciò comporta errori di arrotondamento, di troncamento, di approssimazione da sommarsi a quelli generati dal solutore della PDE discretizzata. Il processo di generazione della griglia di calcolo prevede sempre la presenza nel dominio di calcolo di almeno un 'taglio' necessario ad effettuare la mappatura della regione in una regione rettangolare quale è quella del dominio di calcolo. Secondo il tipo di mappatura del dominio si hanno grid di tipo "O", "C", "H" (comunemente indicati nella letteratura anglosassone come O-type grid, C-type grid, H-type grid). Il tipo di mappatura è dettato dalla maggiore o minore semplicità con cui si riescono a descrivere le geometrie assegnate.

È opportuno ricordare che questi grid nacquero dall'esigenza di risolvere problemi di CFD bidimensionali su profili alari. Si tenga presente che il dominio di calcolo viene mappato in una geometria regolare la cui rappresentazione interna del computer è una matrice multidimensionale su cui poi lavorerà il solutore delle PDE. La presenza di un taglio, e della mappatura effettuata, comporta la duplice esigenza di riscrivere le condizioni al contorno, scegliere la posizione opportuna del taglio (non necessariamente singolo) e assegnare le debite condizioni di continuità lungo i tagli effettuati.

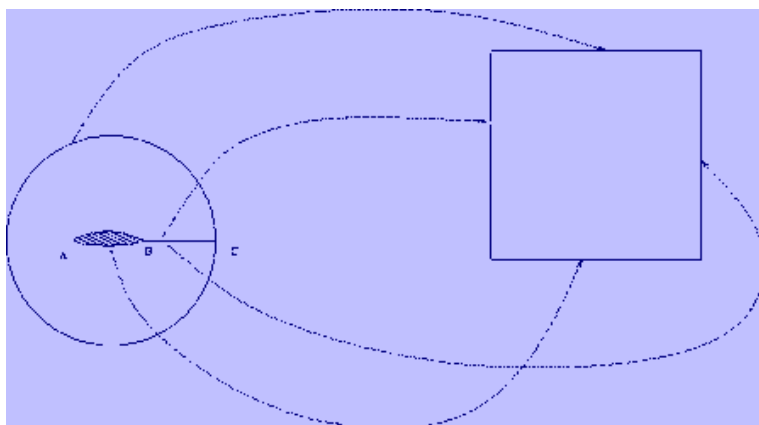


Figura 7

La Figura 7 schematizza la logica di mappatura di un grid di tipo "O" su di un profilo alare (che è un grid polare) mostrando le connessioni fra lo spazio fisico e quello computazionale. Si osservi che il bordo B-A-B del profilo alare diventa il lato inferiore del quadrato, il taglio BC effettuato all'interno del campo viene trasformato nei due lati del quadrato. Il bordo esterno della "O" viene ridotto al lato superiore. Il taglio impone la necessità di eguagliare i flussi sui due lati del quadrato, altrimenti il flusso perde significato fisico.

Le condizioni al contorno di tangenza del flusso sulla superficie del profilo alare devono essere imposte sul lato inferiore del quadrato, mentre le condizioni al contorno all'infinito devono essere imposte sul lato superiore del quadrato.

La Figura 8 mostra un esempio di grid ad O: si noti la maniera in cui vengono imposte le condizioni al contorno e come la mesh venga trasformata passando dallo spazio "fisico" a quello computazionale.

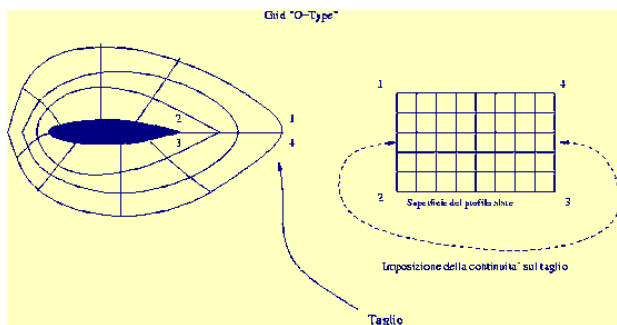


Figura 8

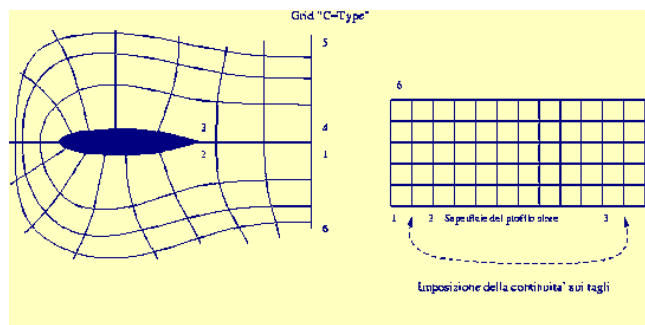


Figura 9

La Figura 9 mostra, invece, la struttura di un grid a C: esso fu sviluppato essenzialmente per mappare profili alari bidimensionali, in seguito fu esteso ai profili tridimensionali.

La mappatura a C genera un grid molto più preciso di quello O-type, di conseguenza la soluzione numerica delle PDE sarà più attendibile, soprattutto nei casi critici in cui ci siano fenomeni di turbolenza e/o onde d'urto. È interessante notare come questa volta la superficie del profilo, nello spazio computazionale, sia circondata dalle superfici corrispondenti al taglio. Ovviamente, è necessario provvedere ad opportune operazioni sulle condizioni al contorno e sulla continuità della soluzione lungo il taglio.

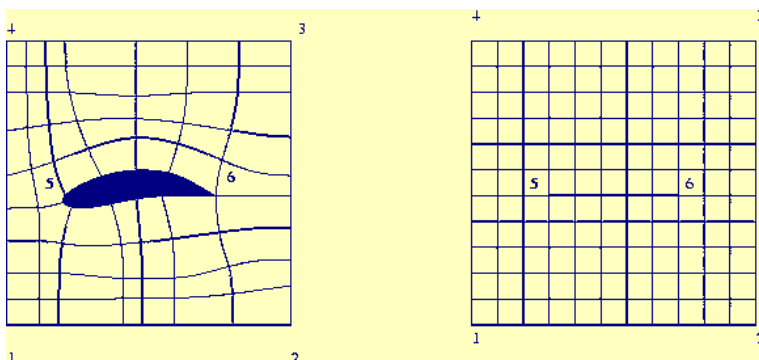


Figura 10

Per ultima si riporta un'immagine di un grid ad H (Figura 10): questa volta il profilo alare cade all'interno dello spazio computazionale, così come i tagli. In questo caso il dominio non è più a connessione semplice, però è possibile descrivere geometrie più complesse in tal modo. Se il flusso è per esempio orizzontale, è necessario associare ad ogni cella anche informazioni sulla normale, altrimenti sarebbe impossibile assegnare le dovute condizioni al contorno. La stessa cosa vale per le condizioni al contorno da applicarsi sulla superficie del profilo alare.

Si accenna al fatto che nel caso di solutori a differenze finite per esempio di tipo forward è necessario introdurre gli "halo points", ossia dei punti fantasma che consentano di utilizzare lo schema anche ai limiti del dominio computazionale (che, si ricorda, non coincidono con i limiti dello spazio fisico), eventualmente facendo saltare il solutore dal bordo ai punti interni. Geometrie intricate possono rendere di fatto impossibile la generazione di una griglia monoblocco. In tali situazioni l'utente del programma di generazione di grid deve ricorrere a strategie multiblocco o di sovrapposizione.

Riguardo a quest'ultimo tipo di grid, vale la pena di ricordare che vengono chiamati Chimera grid. Il nome deriva da un mostro della mitologia Greca poiché i grid ottenuti per sovrapposizione non hanno un aspetto molto bello (all'occhio!).

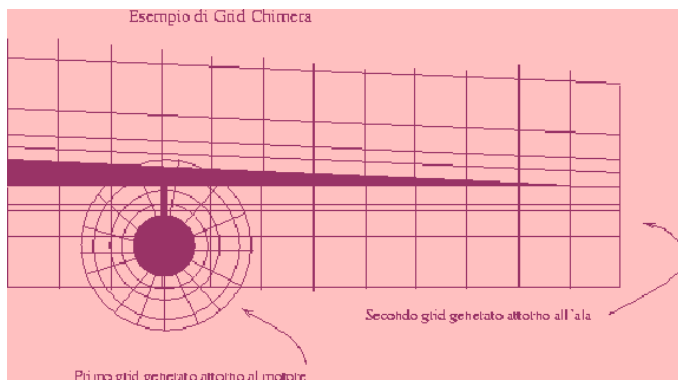


Figura 11

Si veda in proposito la Figura 11, che rappresenta l'applicazione tipica (anzi: quella per cui nacque) di un grid Chimera: è evidente come esso venga ottenuto per sovrapposizione di due mesh costruite su altrettanti corpi. Operando in tal modo è possibile costruire grid strutturati (migliore qualità della soluzione) anche su geometrie che normalmente ne renderebbero impossibile la generazione. Si osservi che la mesh generata attorno al turboreattore è di tipo curvilineo, quella relativa all'ala è Cartesiana, ed inoltre più celle del grid del reattore sono contenute in una del grid dell'ala. Tali situazioni, inevitabili, rappresentano dei punti critici..

La Figura 12 mostra un esempio di Grid Chimera generato sulla superficie di un aereo commerciale, mentre la Figura 13 mostra un particolare di un Grid Chimera Strutturato generato sull'ala di un aereo (entrambe per gentile concessione del NASA Ames Research Center).

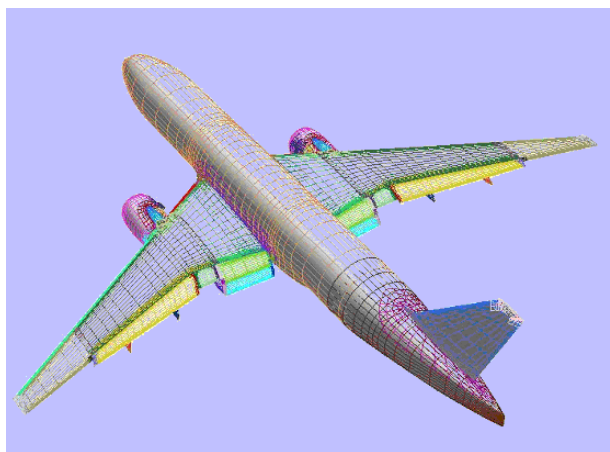


Figura 12

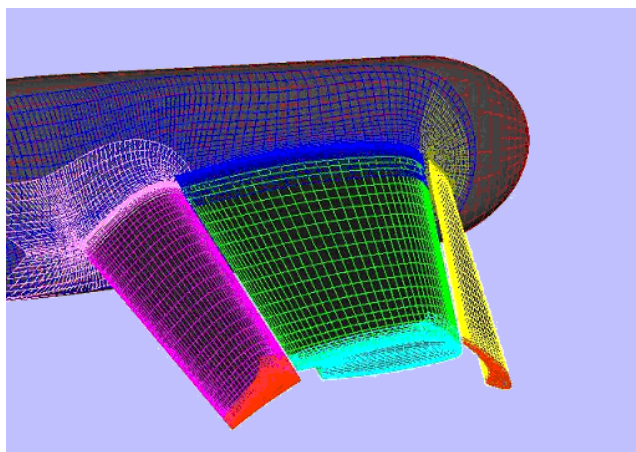


Figura 13

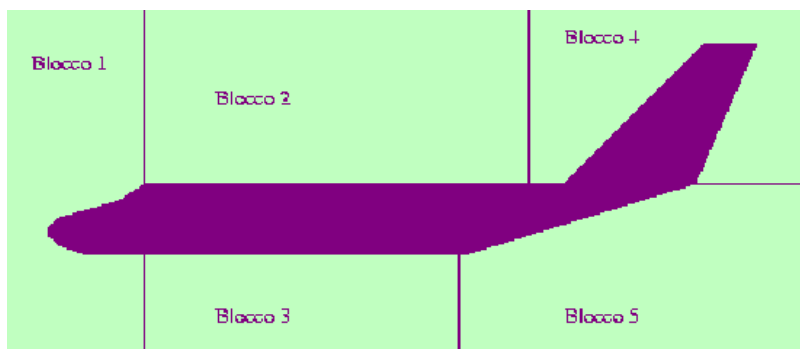


Figura 14

La Figura 14 mostra la logica di suddivisione di un ipotetico grid strutturato multiblocco attorno ad un aereo. Dato che la geometria è complessa, lo spazio viene suddiviso in domini relativamente semplici interconnessi in modo tale da garantire continuità del flusso di informazioni sulle zone di interfaccia. I punti di contorno di un dominio vengono impiegati per generare la griglia in quello adiacente.

In entrambi i tipi di grid, Chimera o multiblocco, è necessario assicurare la corretta continuità del flusso di informazioni da un grid all'altro (variabili fluidodinamiche) e la conservazione delle eventuali condizioni al contorno presenti sui bordi del grid. Negli schemi multiblocco, infatti, ci si può trovare nei seguenti casi:

- perfetta corrispondenza delle celle;
- non corrispondenza delle celle;
- parziale sovrapposizione di celle, corrispondenti e non.

Le Figure 15, 16, 17 che seguono mostrano, rispettivamente, una grid multiblocco con celle corrispondenti, una grid multiblocco a celle non corrispondenti e una grid multiblocco a blocchi sovrapposti.

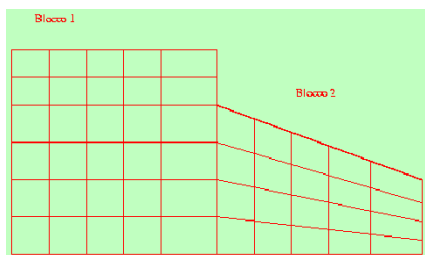


Figura 15

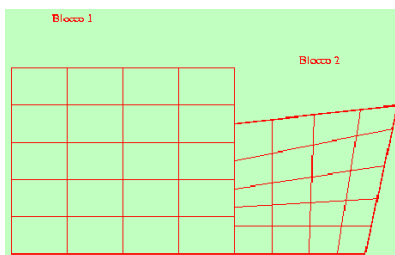


Figura 16

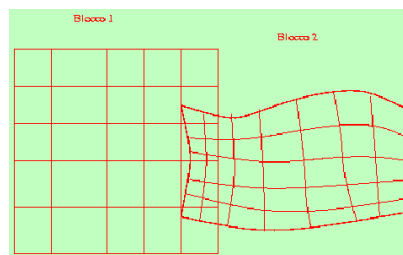


Figura 17

In aggiunta a ciò potrebbe capitare che le celle di due zone siano di diversa natura, per esempio triangolari e quadrate: tale circostanza causa un differente grado di approssimazione della soluzione, oltre a richiedere un solutore capace di girare su diversi tipi di celle. Da un punto di vista fluidodinamico si potrebbero avere discontinuità (inesistenti) della soluzione, in ambito strutturale variazioni di rigidità fittizie (irrigidimento a taglio o a flessione).

Alcune geometrie sono irrealizzabili mediante griglie di tipo strutturato. Si pensi ad un'elica navale, un grid di una zona montagnosa (per calcoli di fluidodinamica delle masse d'aria). In tali casi è necessario ricorrere alle griglie non strutturate che, essendo "svincolate" dalla geometria di partenza, riescono là dove i grid strutturati falliscono, oppure si può ricorrere a griglie ibride.

È possibile adottare strategie miste multiblocco/Chimera in modo tale da poter descrivere geometrie eccezionalmente complesse (per esempio aerei completi, navi), facendo eventualmente ricorso anche a mesh di tipo ibrido, ossia strutturate/non strutturate. È chiaro che tali strategie vengono adottate solamente laddove ci sia convenienza, dato che i tempi di programmazione (e di esecuzione) dei codici di generazione griglie e dei solutori CFD sono, in questi casi, particolarmente lunghi.

La parte fondamentale nella generazione di un grid, quale ne sia il tipo, è la distribuzione di punti sulla superficie da descrivere: essa determina il numero di elementi che compongono il grid (e quindi la pesantezza del grid in termini di spazio occupato in memoria), la qualità della soluzione ottenuta (entro certi limiti, più fitto è il grid migliore è l'approssimazione numerica della PDE), l'allungamento delle celle, la nascita o la morte di celle all'interno del dominio.

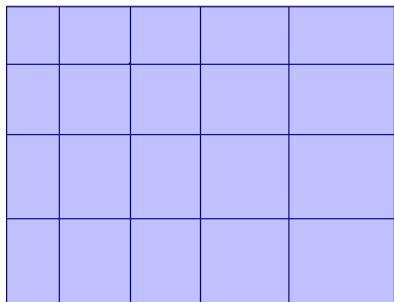


Figura 18

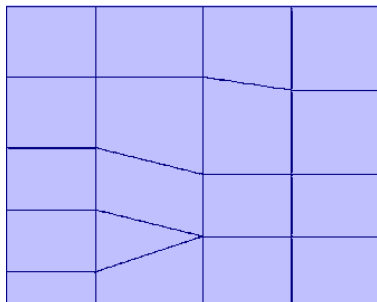


Figura 19

La Figura 18 mostra un esempio di Grid Strutturato Regolare, mentre la Figura 19 mostra un esempio di Distorsione di un Grid Strutturato.

Confrontando la Figura 18 con la Figura 19 appare chiaro il significato di nascita o morte delle celle: dovendo variare il numero di punti lungo una direzione, l'unica possibilità è di aggiungere o togliere alcune celle.

Ciò comporta l'introduzione nella mesh di un certo grado di irregolarità di cui occorre tenere conto nell'analisi dei risultati ottenuti in seguito dalla simulazione numerica. L'esigenza di creare uno spazio computazionale a partire da uno spazio fisico infatti, comporta il ricorso ad uno Jacobiano (Figura 20).

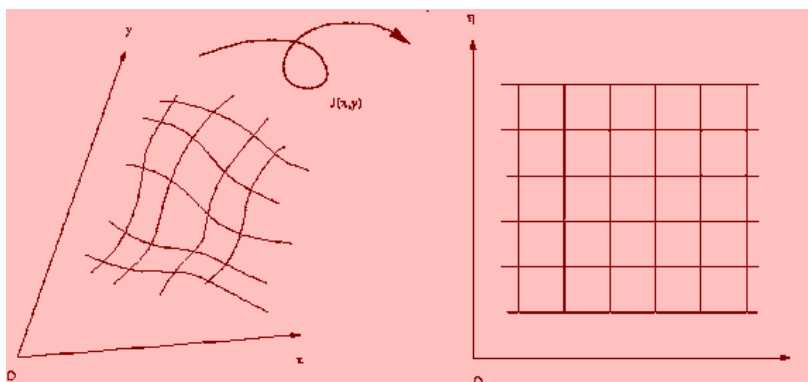


Figura 20

Tale Jacobiano, nel caso di celle allungate (rapporto di allungamento maggiore di circa 8) o aventi angoli eccessivamente acuti od ottusi, può diventare minore o uguale a zero. Questo, da un punto di vista numerico, equivale ad affermare che il volume (area) della cella considerata è nullo o negativo, e ciò è chiaramente impossibile. Nei punti del dominio in cui ciò avviene si potrebbe avere mancata stallo del solutore (loop infiniti) o convergenza verso valori assurdi.

È perciò necessario controllare senza troppa fretta la qualità della mesh generata. In linea di massima una griglia è soddisfacente se:

- È una valida mappatura del dominio assegnato (volumi e/o aree positive),
- Il rapporto di allungamento delle celle è contenuto (di regola inferiore a dieci),
- È una rappresentazione sufficientemente 'levigata' della geometria considerata (per esempio: una circonferenza deve essere descritta da un adeguato numero di segmenti, sì da sembrare una circonferenza e non un esagono...),
- Gli angoli delle celle sono vicini a 60° nei grid non strutturati, compresi fra 45° e 135° nel caso di grid strutturati.

GENERAZIONE DI GRIGLIE STRUTTURATE

La generazione di griglie strutturate può essere ottenuta per via diretta o indiretta. La generazione per via diretta è ottenuta mediante:

- Generazione univariate (leggere in inglese);
- Generazione bivariate, detta anche TFI, la quale a sua volta può essere:
 - Lineare
 - Di Hermite
 - Di Bezier
 - Di tipo Spline
 - Una combinazione delle precedenti strategie.

La generazione di tipo indiretto è conseguita tramite:

- Risoluzione di PDE di tipo:
 - Ellittico;
 - Iperbolico;
 - Parabolico;
- Tecniche variazionali.

È ovvio che richiedendo questi ultimi metodi la soluzione di una equazione differenziale, la generazione di griglie, benché di qualità superiore rispetto a quella ottenuta tramite i metodi algebrici diretti, richiede un tempo di elaborazione computazionale maggior.

Interpolazioni Univariazionali

Le *interpolazioni univariazionali* sono semplici da implementare e da usare, ma non consentono un controllo ottimale della distribuzione dei punti all'interno del dominio di calcolo. Un migliore comportamento è ottenuto mediante l'adozione di *B-splines* o *curve di Bezier*, che permettono controllare sia i punti interni che quelli sul bordo.

Interpolazioni Transfinite

Le *interpolazioni transfinite* (d'ora in poi: TFI, da *Trans Finite Interpolation*) sono il sistema comunemente adoperato per generare griglie di tipo strutturato evitando i problemi causati dalle interpolazioni univariazionali. Il concetto base è quello di avere una curva madre da cui partono famiglie di curve simili che si deformano con continuità fino a assumere, all'estremità opposta del dominio, la forma della curva che ne definisce il bordo. Al variare del tipo di curva utilizzata per generare le TFI si hanno le diverse tipologie. L'interpolazione lineare prevede l'uso di polinomi interpolatori del primo ordine, quella di Hermite impone l'impiego di polinomi di Hermite e così via per le Spline e le curve di Bezier.

Ciò che cambia da una tipologia all'altra di TFI è la capacità di descrivere domini geometricamente più complessi passando dai polinomi del primo ordine alle curve di Bezier. Le TFI generate mediante polinomi di Hermite sono caratterizzate dalla ortogonalità: una mesh possiede tale importante proprietà se le curve che la definiscono partono e arrivano ortogonali agli estremi del grid.

Una TFI è definita da più funzioni: 2 in 2D, 3 in 3D. Ogni funzione, monodimensionale, prende il nome di proiettore, e la somma Booleana dei proiettori origina la TFI secondo la formula:

$$(3) \quad F_1 + F_2 + F_3 = F_1 + F_2 + F_3 - F_1F_2 - F_2F_3 - F_3F_1 + F_1F_2F_3$$

La (3) è l'espressione di una TFI tridimensionale: si osservi che è generata da proiettori F_i monodimensionali. Nonostante siano nate per ridurre il problema dell'overlapping anche le TFI, se impiegate in geometrie aventi una dimensione che varia fortemente (si immaginino geometrie a forma di tromba, imbuto), possono originare tale problema. Per ovviare a tali inconvenienti si è passati alla generazione indiretta dei grid.

Generazione di Tipo Indiretto

La generazione di grid mediante tecniche dirette è particolarmente veloce, ma non consente di ottenere grid particolarmente affinati (nel senso suddetto). Il ricorso a tecniche indirette è allora obbligatorio, nonostante le maggiori difficoltà di implementazione ed i tempi di generazione più lunghi.

L'uso di equazioni differenziali alle derivate parziali risulta utile in tal senso, ma rende necessario l'impiego di tecniche numeriche per risolverle, dato che non sono equazioni lineari. Così facendo, per risolvere una equazione differenziale se ne solve un'altra, per giunta non lineare, per generare il grid.

Un primo esempio di grid ottenuto mediante PDE è dato dalle equazioni di tipo ellittico, come l'equazione di Laplace. Considerato un caso bidimensionale, se x ed h sono le coordinate dello spazio computazionale, le PDE che governano la generazione della mesh sono:

$$(4) \quad \begin{aligned} \nabla^2 \xi &= 0 \\ \nabla^2 \eta &= 0 \end{aligned}$$

Thompson ha elaborato un metodo più accurato del precedente tramite la soluzione dell'equazione di Poisson. Tale metodo consente un maggior controllo della distribuzione di punti sulla geometria da parte dell'operatore. In tal caso le (4) assumono espressione:

$$(5) \quad \begin{aligned} \nabla^2 \xi &= (\nabla \xi \cdot \nabla \xi) \Phi \\ \nabla^2 \eta &= (\nabla \eta \cdot \nabla \eta) \Psi \end{aligned}$$

Nella (5) le funzioni Φ e Ψ sono dette funzioni di controllo (alle volte indicate con p e q nella letteratura tecnica) in quanto consentono di gestire la dislocazione dei punti sul bordo della geometria da descrivere permettendo, inoltre, di gestire l'ortogonalità del grid. La loro scelta non è facile, poiché da esse dipende la capacità del generatore di griglie di evitare l'overlapping (Figura 21 e Figura 22). Il sistema di equazioni risultante dalla generazione ellittica viene risolto mediante tecniche *SOR* (*Successive Over Relaxation*).

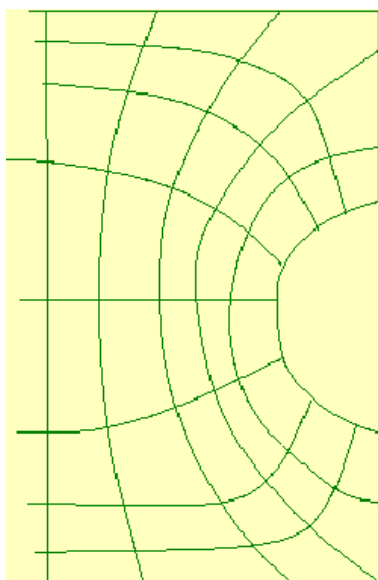


Figura 21

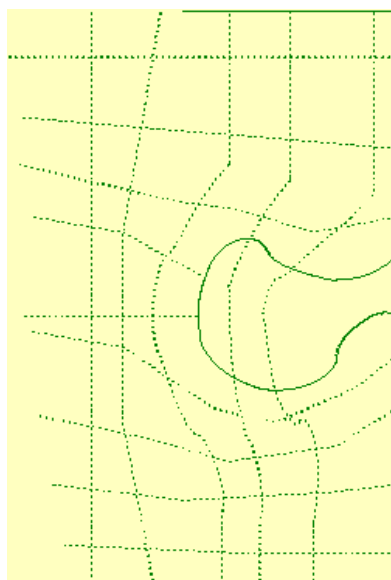


Figura 22

Una particolarità dell'affinamento ellittico è che le linee di grid vengono attratte verso i vertici convessi, allontanate da quelli concavi. Ciò porta ad un vuoto nel caso, per esempio, di un angolo interno di un rettangolo, ad una sovrapposizione di linee se si sta creando il grid dalla parte esterna del medesimo.

I metodi di tipo ellittico sono interessanti fintantoché non si guardi al tempo richiesto per generare una griglia, soprattutto nei sistemi tridimensionali. Il consumo di tempo è causato dalla necessità di risolvere le equazioni in maniera reiterata. Nakamura (1982) risolse il problema "parabolizzando" le equazioni ellittiche di generazione delle griglie.

In tal modo (si omettono qua le formule) si ottiene un sistema di generazione di mesh più veloce di quello ellittico, ma anche meno regolare. Un ulteriore vantaggio degli schemi parabolici è che sono comodi da usare per i grid multiblocco. Da un punto di vista matematico gli schemi parabolici possono essere interpretati come una approssimazione di quelli ellittici.

I metodi di tipo iperbolico nascono per risolvere un problema di ambito squisitamente CFD: accade spesso, soprattutto nei problemi di aerodinamica esterna, che la zona di interesse sia ristretta ad una regione vicina al corpo in esame. Ciò che accade a grande distanza dal corpo ("infinito") non è generalmente di interesse pratico. Tuttavia è necessario provvedere alla generazione di mesh anche laddove il calcolo non sia di interesse perché bisogna provvedere le opportune condizioni al contorno. In questi casi le tecniche iperboliche vengono incontro consentendo di generare grid fitti nelle vicinanze del corpo e larghi a "grande" distanza da esso. Il risultato è una griglia più leggera e tempi di calcolo ridotti rispetto ai precedenti casi.

La tecnica iperbolica è molto efficiente poiché il grid è generato in un singolo passo e la qualità della griglia è elevata per corpi dalle geometrie piuttosto regolari, inoltre viene eliminata la necessità di definire punti sul bordo esterno del dominio computazionale. Tale operazione, infatti, può risultare dispendiosa in termini di tempo qualora si debbano definire molti punti sulla superficie fisica del corpo in esame (si tenga presente che la distribuzione di punti sulle superfici, quali che siano - controlla la mappatura dei punti interni della regione di spazio oggetto della mesh) soprattutto allorché si usino le TFI. Anche questa tecnica, come tutte le altre, è affetta da problemi: le irregolarità geometriche presenti sulla superficie del corpo possono propagarsi nello spazio esterno (cioè nel campo di moto del fluido), inoltre zone concave possono causare sovrapposizione di celle (con gli ovvi inconvenienti) e per finire il metodo non può adattare il grid ad un bordo

esterno definito. Si confrontino, a tal proposito, la Figura 21 (Grid Omogeneo in una zona concava) e la Figura 22 (Grid sovrapposto in zona concava): appare chiara la qualità maggiore del primo rispetto al secondo.
 La Tabella 1 riassume il confronto fra le varie tipologie di grid:

| Caratteristica | Grid Algebrico | Affinamento Laplace | Affinamento Poisson |
|---------------------------------|----------------|---------------------|----------------------|
| Mappatura valida | Non Garantito | Garantito | Controllato da p e q |
| Levigatezza | Non Garantito | Garantito | Garantito |
| Ortogonalità | Non Garantito | Non Garantito | Controllato da p e q |
| Rapporto di allungamento | Controllabile | Non Controllabile | Controllato da p e q |

Tabella 1

La generazione di mesh tramite soluzione di PDE richiede in ogni caso elevati tempi di calcolo se paragonati a quelli tipici di altri metodi algebrici. Il metodo iperbolico è più rapido di quello ellittico, il parabolico si colloca in mezzo ai due.
 A titolo di esempio si riportano di seguito alcune immagini descriventi la generazione di un grid strutturato attorno ad un elica ed attorno alla poppa di una nave. L'elica rappresenta un caso particolarmente complesso da risolvere a causa dello svergolamento delle pale e a causa della struttura della zona poppiera di una nave.

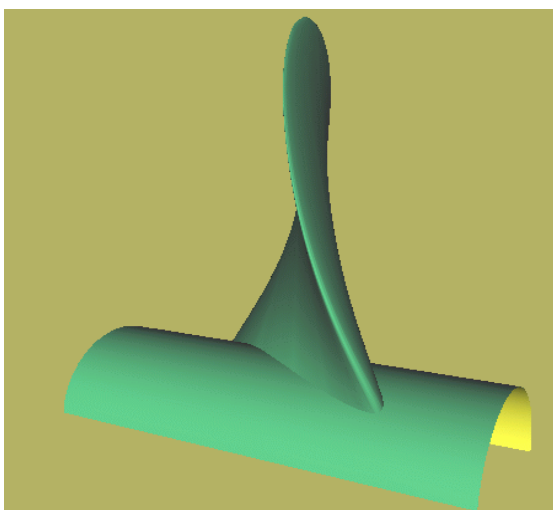


Figura 23

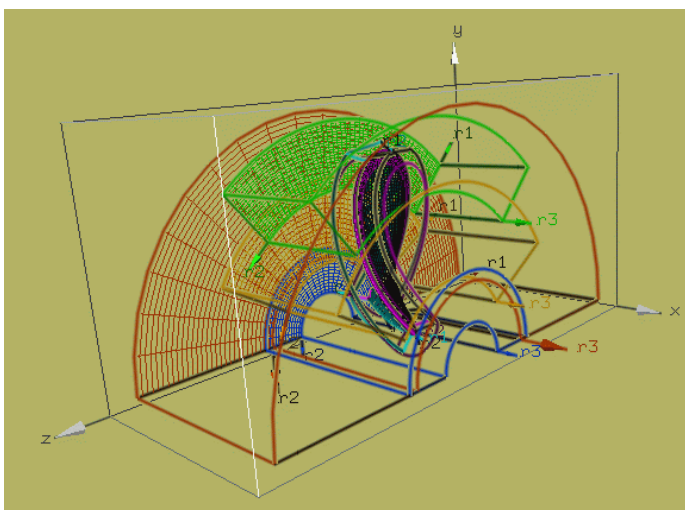


Figura 24

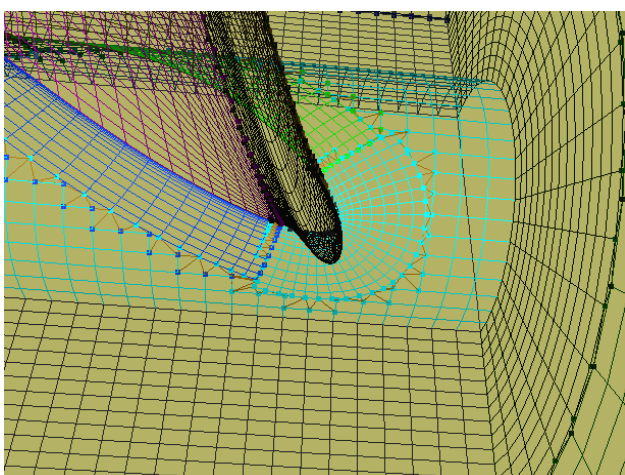


Figura 25

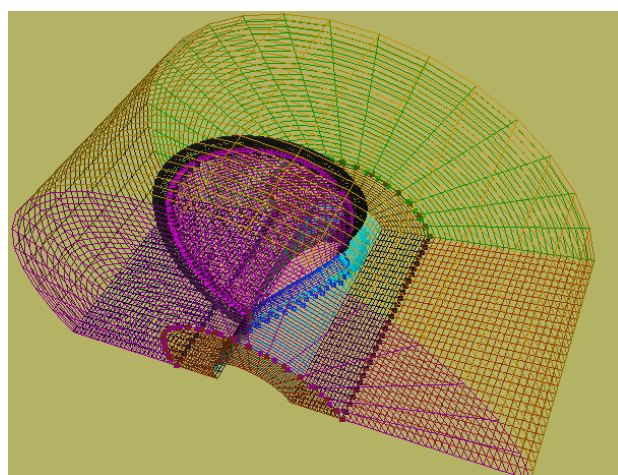


Figura 26

Si osservi il modo in cui viene suddivisa la regione di spazio attorno all'elica e le transizioni nelle mesh alla radice della pala (Figura 25): per unire i vari blocchi del grid si è resa necessaria l'adozione di elementi di transizione triangolari. Tali elementi potrebbero essere fonte di mancata convergenza o minore approssimazione della soluzione delle PDE.

La Figura 23 rappresenta la superficie fisica di un'elica navale (per gentile concessione del Dr. Anders Petersson), in questo caso dotata di una sola pala (in effetti è un'elica piuttosto "didattica"...). Tale elica è stata ottenuta generando innanzitutto delle spline, poi su tali spline è stata definita la distribuzione di punti da cui è stato generato il grid.

Quindi, a partire dai suddetti punti è stata generata la mesh in tutto il campo. Si fa presente che questo grid è misto, ossia Strutturato e Non Strutturato. Le Figure 24, 25 e 26 (per gentile concessione del Dr. Anders Petersson) mostrano, rispettivamente, la spline di generazione del grid dell'elica, le superfici di base per la generazione del Grid e il grid completo attorno all'elica.

GRIGLIE NON STRUTTURATE

In tempi relativamente recenti sono state sviluppate le griglie di tipo non strutturato che consentono di descrivere geometrie complesse costi computazionale e operativo (in termini di tempo speso per la generazione della griglia) drasticamente ridotti rispetto a quanto necessario per sviluppare un grid strutturato. Si accenna al fatto che i concetti base della generazione di griglie non strutturate furono esposti da Dirichlet in una sua pubblicazione nel 1850.

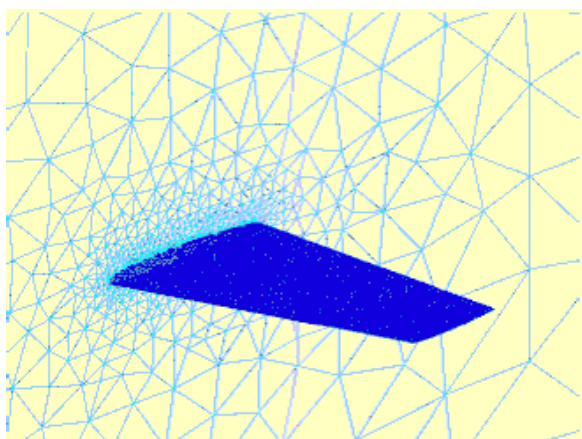


Figura 27

Il principale vantaggio di una griglia non strutturata è che è sempre possibile suddividere un dominio in elementi triangolari, nel caso di problemi bidimensionali, o tetraedrici, nel caso di problemi 3D. È inoltre possibile variare la densità della griglia conformemente al gradiente della soluzione: la mesh viene infittita solo dove serve, per esempio nelle zone in cui si modella la turbolenza oppure laddove avvengono i fenomeni fisici di maggior interesse. Ciò non sempre è possibile quando si adottano grid strutturati a causa delle difficoltà insite nella descrizione di geometrie complesse. La Figura 27 mostra il grid non strutturato di un profilo alare ONERA M6.

I grid non strutturati presentano, inoltre, la possibilità di automatizzare il processo di generazione con i considerevoli vantaggi pratici che ne derivano. Allo stato attuale della tecnica tale possibilità è di fatto concretata per alcuni preprocessori di uso commerciale.

Lo svantaggio principale dei grid non strutturati -almeno per chi debba programmare un codice- è rappresentato dalla necessità di includere esplicitamente i dati riguardanti la connettività. In una griglia di tipo strutturato l'elemento i -esimo è adiacente all' $(i+1)$ -esimo mentre ciò non avviene in una non strutturata, di conseguenza è necessario produrre sia informazioni geometriche sulla mesh, sia informazioni sulla connettività dei nodi. Per esempio:

| ELEMENTO | NODO 1 | NODO 2 | NODO 3 |
|----------|--------|--------|--------|
| 20 | 48 | 104 | 700 |
| 21 | 2 | 51 | 16 |

È evidente che le condizioni al contorno delle singole celle richiedono un trattamento particolare: nodi vicini nello spazio fisico diventano distanti nello spazio computazionale con ovvia complicazione degli algoritmi risolutivi. Inoltre la parallelizzazione dei codici e la loro vettorizzazione diventano più complesse.

È da osservare che nel caso di griglie di tipo strutturato le informazioni relative alla connettività vengono memorizzate secondo una logica a blocchi piuttosto che a celle. La memorizzazione di un numero elevato di informazioni porta, nel caso delle mesh non strutturate, a file di dimensioni maggiori rispetto a quelle relative al caso strutturato.

Vicinanza di Voronoi – Triangolazione di Delaunay

Si consideri un insieme di punti non allineati e distribuiti in modo tale che non possano giacere su di una circonferenza. L'insieme di punti che si trova ad essere il più vicino possibile ad un vertice v_i rispetto agli insiemi ad esso vicini, è detto essere in *Vicinanza di Voronoi* (pronuncia: foronoi, dal matematico Russo che definì, nel 1908, tale proprietà dei punti). La suddivisione di un dominio nel senso di Voronoi (da alcuni definita come suddivisione di Dirichlet) è costituita da poligoni di Voronoi.

La *Triangolazione di Delaunay* (pronuncia: delonè) è la base per la costruzione di grid non strutturati, ed è imperniata sulla Suddivisione di Voronoi, essendone praticamente il complemento (Figura 28). Laddove sia applicata, la triangolazione di Delaunay produce grid che godono delle seguenti proprietà:

- 1) dato un insieme di punti, la triangolazione del dominio è univoca;
- 2) per un assegnato insieme di punti, la triangolazione di Delaunay produce i triangoli il più equilateri possibile;
- 3) la generazione dei punti della griglia e la triangolazione sono due processi separati;

La Triangolazione di Delaunay gode delle seguenti proprietà:

- dato un punto P del dominio ed un cerchio di raggio r centrato in esso, non esistono punti giacenti nelle vicinanze di P che siano contenuti all'interno del cerchio;
- la triangolazione di Delaunay è quella che massimizza il minimo angolo del quadrilatero convesso generato da due triangoli aventi un lato in comune. Tale proprietà è detta proprietà equiangola.

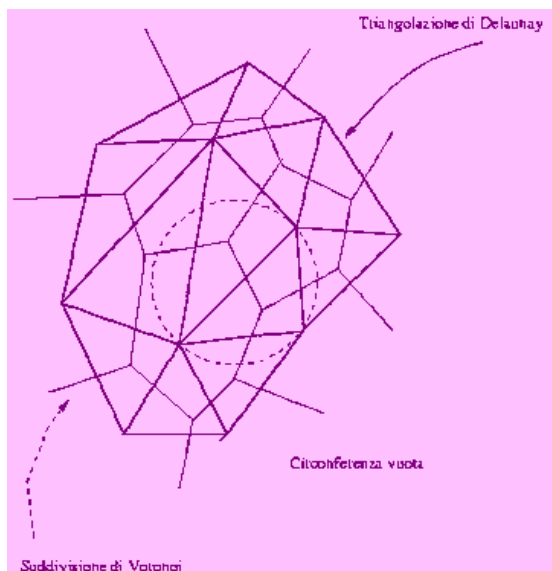


Figura 28

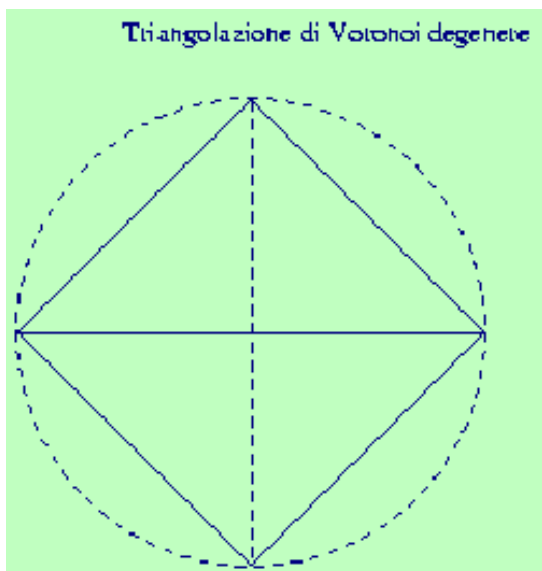


Figura 29

La prima proprietà è trasferibile a mesh tridimensionali laddove al triangolo si sostituisca il tetraedro ed al cerchio la sfera. La seconda è caratteristica dei grid bidimensionali. Per un dato punto la triangolazione di Delaunay è sempre unica ad eccezione del caso degenerato illustrato in Figura 29. Tale configurazione è, ovviamente, da evitare in quanto formerebbe un punto di singolarità all'interno del dominio di calcolo. Esistono vari sistemi per generare griglie non strutturate. Uno degli algoritmi più usati è il Bowyer-Watson (1981), applicabile a sistemi bi e tri dimensionali. L'algoritmo procede come di seguito descritto:

1. viene generato il triangolo iniziale;
2. viene aggiunto un punto in maniera casuale;
3. vengono ricercati triangoli i cui circocentri contengono il nuovo punto;
4. i triangoli vengono cancellati creando delle aree sempre convesse;
5. tutti i punti che risiedono nelle vicinanze del punto in esame vengono congiunti ad esso creando in tal modo una nuova famiglia di triangoli;
6. la struttura dei dati viene rigenerata e si torna al punto (2).

Se il dominio di calcolo è costituito da N nodi, nel caso peggiore l'algoritmo richiede un tempo di generazione che è ordine $O(N^2)$, nel caso migliore è quasi $O(N)$. È possibile affinare l'algoritmo in altri modi che non vengono qui analizzati per evitare di appesantire troppo l'articolo. La triangolazione di Delaunay è la più veloce possibile, ma può produrre, particolarmente con elementi tetraedrici in domini 3D, triangoli non idonei al calcolo, per esempio triangoli con angoli troppo piccoli. È necessario, quindi, provvedere ad un controllo sulla geometria della mesh al termine della generazione della stessa. Tale controllo è effettuato in maniera automatica dai programmi di generazione di grid, e prevede di solito almeno due spazzate del dominio: una prima di generazione, un'altra di affinamento degli elementi generati. Normalmente è previsto un input dall'utente che può decidere quanti punti disporre sulle superfici iniziali, impostare limiti di accettabilità dei triangoli (di solito angoli compresi fra 30° e 120°) oppure decidere la rapidità con cui la mesh deve crescere a partire dalle superfici della geometria in esame. La triangolazione di Delaunay è caratterizzata dalle seguenti proprietà:

- può essere resa molto efficiente;
- garantisce una distribuzione di punti regolare;
- è abbastanza semplice da implementare;
- il piazzamento di punti per creare i circocentri causa elementi di bassa qualità in corrispondenza dei bordi del dominio;
- i grid tridimensionali, non essendo garantita l'equiangolarità dei triangoli componenti il tetraedro, possono essere degeneri;
- è necessario garantire una valida connettività di Delaunay in tutto il dominio;
- l'affinamento del grid in corrispondenza dei bordi del dominio può essere difficoltosa.

Metodo ad Avanzamento del Fronte

Il metodo di avanzamento di fronte è una tecnica alternativa per sviluppare griglie non strutturate in cui i punti vengono generati e connessi allo stesso tempo. Essa è essenzialmente dovuta a Peraire (1987), il quale elaborò anche la strategia per griglie 3D nel 1990. La generazione della mesh avviene partendo da segmenti in 2D e triangoli in 3D: vengono creati dei punti che consentono di espandere la mesh nello spazio finché non si è riempito il dominio di interesse con triangoli o tetraedri.

La generazione del grid è controllata in maniera tale da garantire una certa regolarità delle geometrie generate. Il metodo di avanzamento del fronte è caratterizzato dalle seguenti proprietà:

- gli elementi generati sono di elevata qualità;
- gli elementi sui bordi sono automaticamente affinati;
- la procedura è localizzata, si che è possibile, qualora necessario, procedere ad un affinamento locale della mesh;
- l'efficienza è inferiore rispetto al metodo di Delaunay;
- speciali strutture di dati sono richieste per impostare il fronte ed ottenere una buona velocità di avanzamento del fronte stesso;
- i fronti potrebbero originare collisioni in una zona non triangolabile.

Il problema da risolvere è il seguente: suddividere un dominio in sottodomini elementari aventi forme triangolari o tetraedriche che abbiano al più in comune un punto, un lato o una faccia (domini consistenti). L'algoritmo impiegato in 2D è il seguente:

1. le curve che definiscono il contorno dei corpi vengono discretizzate;
2. vengono piazzati dei nodi su tali curve e quindi creati dei segmenti unenti i nodi contigui aventi dimensione congrua con la densità di mesh desiderata. L'insieme dei nodi e dei segmenti costituisce, in un dato istante, il fronte di avanzamento della griglia. La Figura 30 mostra la schematizzazione del Metodo di Avanzamento di Fronte per la generazione di grid;
3. a partire dal fronte vengono generati i triangoli. La generazione può avvenire o per creazione di nuovi punti, o per connessione di punti già esistenti;
4. la procedura viene reiterata finché il dominio non è stato suddiviso completamente. Durante il processo "di crescita" della mesh è possibile variare l'altezza dei triangoli in modo tale da sfolire la griglia a mano a mano che ci si allontana dal bordo delle curve madri. Si veda a tal proposito la Figura 31 che mostra un esempio di grid non strutturato del profilo di Joukowski.

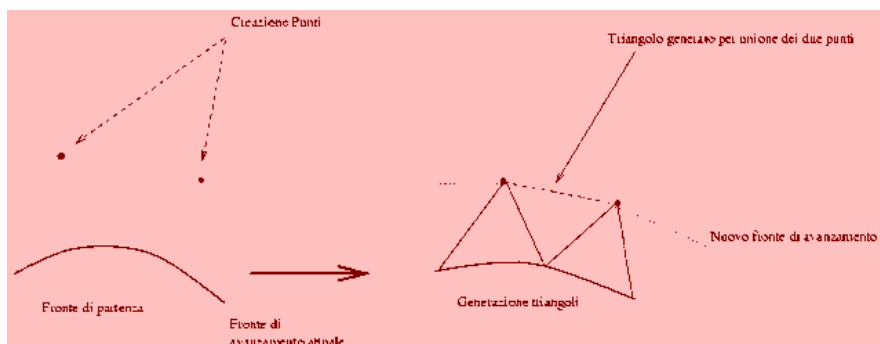


Figura 30

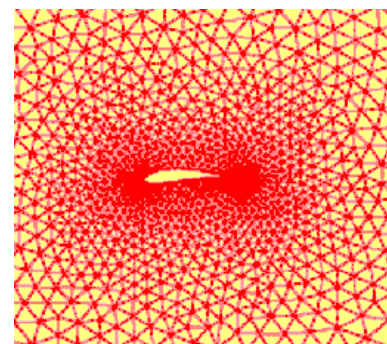


Figura 31

Nel caso di strutture tridimensionali si sostituiranno i tetraedri ai triangoli e le superfici (triangolari) ai segmenti. È possibile combinare la triangolazione di Delaunay con il metodo di avanzamento del fronte onde ottenere strategie che combinino i vantaggi di entrambi i metodi.

Grid di Tipo Ibrido

Nelle applicazioni CFD interessa in genere risolvere il campo di moto in una zona relativamente limitata di spazio, ovvero lo strato limite. Dato che i fenomeni di interazione viscosa fra fluido e materia sono confinati all'interno dello strato limite, la conoscenza di ciò che avviene al di fuori di esso, fatto salvo il caso di separazione dello strato limite, può non richiedere un grid eccessivamente fitto, a tutto vantaggio della semplicità di generazione e della rapidità di calcolo.

La soluzione risulta in genere essere più accurata se calcolata su grid strutturati, le griglie di tipo non strutturato comportano soluzioni in genere più approssimate.

Da un punto di vista di generazione delle griglie, la soluzione non strutturata è certamente quella più semplice da adottare sia in termini di tempo computazionale di generazione del grid sia in termini di ore-uomo necessarie a impostare la mesh. La generazione di mesh strutturate, si è visto, comporta una serie di problemi laddove le superfici dei corpi da descrivere non siano sufficientemente lisce.

Si è pensato, quindi, di ideare delle *mesh di tipo ibrido*, ossia strutturate e non strutturate, per usufruire dei vantaggi di entrambe le tipologie: da un lato, le griglie strutturate garantiscono una maggiore correttezza del risultato ottenuto dalla simulazione, dall'altro, i grid non strutturati facilitano la generazione della mesh nel campo, oltre ad accelerare la soluzione. In questa maniera si ottengono risultati soddisfacenti in termini di tempi di calcolo e precisione.

È necessario, però, fissare l'attenzione su due particolari:

1. è necessario disporre di un solutore che abbia capacità di calcolo sia su grid strutturati che su grid non strutturati;
2. la mesh strutturata deve estendersi per tutto lo strato limite.

Il punto (1) è un vincolo notevole per chi debba codificare il solutore. Il punto (2) evidenzia la necessità di un'attenta analisi del flusso e della geometria prima della generazione della griglia. Le griglie ibride sono correntemente impiegate nei problemi CFD.

Metodi Multigrid

I Metodi Multigrid videro luce nel 1964 ad opera del matematico Fedorenko (Russia), ma la prima applicazione pratica fu eseguita nel 1977 da Brandt (Israele).

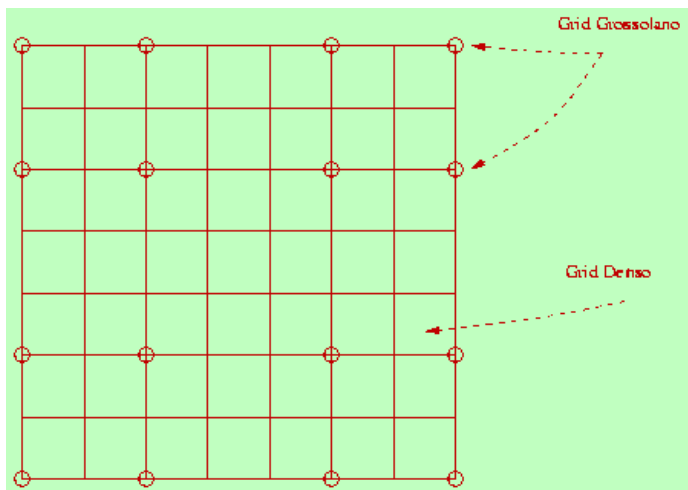


Figura 32

All'inizio degli anni '80 ci fu nella comunità scientifica internazionale un vivo interesse nei confronti di tali metodi, e si incominciò a produrre i primi studi estensivi sull'argomento. A titolo di curiosità si rammenta che negli anni '60, nell'ex URSS, i calcoli di fluidodinamica numerica venivano svolti in grandi capannoni dove centinaia di persone eseguivano un pezzetto di calcolo e poi lo passavano al vicino di banco, il quale compieva un altro step e così via. Si rese necessario, quindi, escogitare un metodo per accelerare la soluzione.

I metodi multigrid non consistono tanto in una tecnica di generazione di griglie quanto, piuttosto, nell'uso che se ne fa. In tale tecnica, infatti, vengono adottati due grid, uno fine e l'altro grossolano, vedasi Figura 32, in maniera tale da accelerare la convergenza della soluzione su quello grezzo ed affinarne il valore su quello denso.

Un esempio classico di tecniche multigrid è il cosiddetto *V-cycle* (Figura 33), ossia *ciclo a V*, che nei suoi tratti essenziali consta delle seguenti fasi:

- il residuo viene ridotto al minimo sul grid grossolano (h nella Figura 33);
- il residuo viene ristretto sul grid grossolano;
- i precedenti passi vengono ripetuti in maniera ricorsiva fino a che il grid a maglia più larga non viene raggiunto ($16h$ nella Figura 33);
- il campo di moto retto dalle PDE viene a questo punto risolto (vertice della V);
- si interpola ricorsivamente sui vari grid finché non si è raggiunta nuovamente la mesh di dimensione media h ;
- il processo viene reiterato finché non si sia ottenuta la convergenza della soluzione.

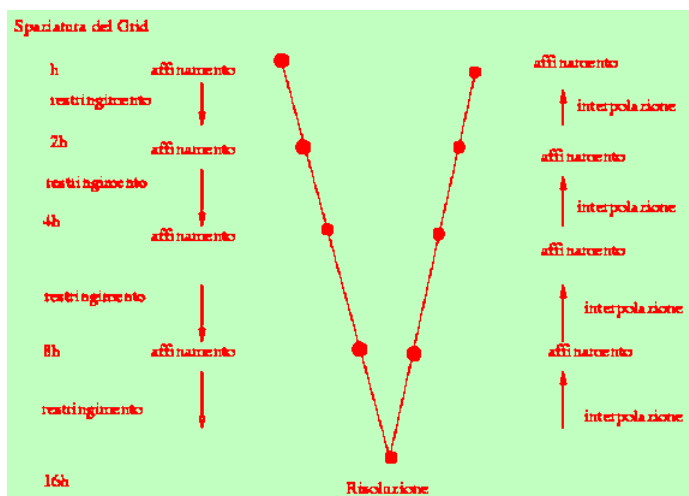


Figura 33

Il detrimento di un metodo multigrid, chiaramente, è rappresentato dalla mole di dati da gestire nonché dalla complessità del sistema solutore-grid da utilizzare. In generale si preferisce impiegare grid strutturati, anche se non mancano esempi di applicazioni con griglie non strutturate. Il prezzo da pagare nel secondo caso è di una ancora maggiore difficoltà di programmazione. Esistono anche altri schemi multigrid, ad esempio il W-cycle. Per maggiori informazioni si rimanda a testi specifici sull'argomento. I metodi multigrid lavorano bene con certi tipi di equazioni, in particolare con PDE di tipo ellittico (lineari e non) definite in un dominio privo di punti singolari. Per PDE non ellittiche i metodi multigrid sono di alcuni ordini di grandezza più lenti.

Da un punto di vista di parallelizzazione del software, i metodi multigrid sono facilmente implementabili ma richiedono un flusso di dati maggiore rispetto ad altri tipi di solutori.

Nonostante ciò il metodo multigrid è impiegato laddove il trade-off fra complessità dei solutori, tempi di calcolo, costi hardware e pesantezza dei grid è tale da consigliarne l'uso. Conviene qua sottolineare un particolare: non esistono teorie matematicamente definite circa l'uso dei multigrid, la corrente ricerca è volta proprio a definire in maniera deterministica criteri di codifica e d'uso delle tecniche multigrid da anteporre agli attuali criteri sostanzialmente euristici.

Per trarre il maggior vantaggio possibile dall'uso dei multigrid è necessario effettuare una ottimizzazione spinta dei codici. Tale ottimizzazione è, purtroppo, dipendente dal tipo di problema che si vuole risolvere nonché dal tipo di solutore adottato, e richiede quindi personale altamente specializzato. Sono questi gli handicap principali dei metodi multigrid.

Affinamento dei Grid

L'affinamento dei reticoli di calcolo è noto come *h-refinement*. In realtà più che di affinamento di un grid, che farebbe pensare ad un miglioramento generalizzato della griglia, questo metodo consiste nell'inserzione di punti nella griglia laddove sia necessario incrementare l'accuratezza della soluzione, aumentando in tal modo il numero di elementi e diminuendone al contempo la dimensione media a livello locale. L'affinamento è relativamente facile da ottenere nel caso di grid di tipo non strutturato, mentre è decisamente più complesso da realizzare per le griglie strutturate, e lo è in ogni caso quando si abbiano mesh multiblocco. Uno dei più comuni inconvenienti generati dagli affinamenti dei grid è quello del peggioramento del rapporto di allungamento delle celle. Dato che i punti vengono aggiunti a mesh già creata, diventa difficile rigenerarla in modo tale da garantire la conservazione di tale parametro.

Grid Adattivi

I *grid adattivi* sono dei grid in cui la densità di griglia viene aumentata laddove il gradiente della soluzione è forte. Per esempio la mesh viene infittita nelle zone dello strato limite, oppure nelle vicinanze di un urto sonico, o ancora dove ci sia turbolenza. In tal modo si migliora la qualità della soluzione nei punti in cui interessa e si velocizza il processo nelle regioni di spazio in cui le variabili fluidodinamiche non subiscono forti variazioni. In effetti, più che una tecnica di generazione di reticoli questa è una tecnica di deformazione dei grid mirata ad un duplice scopo: da un lato la velocità di calcolo, dall'altro la precisione della soluzione (in genere queste due desiderabili qualità sono antitetiche).

CONCLUSIONI

A conclusione, si riporta nella Tabella 2 la sintesi delle caratteristiche dei grid, dalla quale è possibile evincere la maggiore o minore convenienza di un grid rispetto ad un altro.

| QUALITÀ | Strutturato | Non strutturato | Chimera | Grid di tipo misto |
|--|-------------|-----------------|----------|--------------------|
| Flessibilità geometrica | minore | maggiore | maggiore | maggiore |
| Capacità di adattamento del grid | minore | maggiore | minore | maggiore |
| Idoneità al calcolo di flussi viscosi (celle ad alto rapporto di allungamento) | maggiore | minore | maggiore | maggiore |
| grid deformabili | minore | maggiore | maggiore | maggiore |
| quantità di memoria richiesta | maggiore | minore | maggiore | incerta |
| potenza CPU richiesta | maggiore | minore | maggiore | incerta |

Tabella 2

É da tener presente, durante la generazione della mesh, che:

- più alto è il numero di celle, migliore è la soluzione;
- più alto è il numero di celle, più tempo sarà richiesto al solutore per convergere;
- non necessariamente aumentando il numero di celle migliora la soluzione, almeno localmente. Si potrebbero infatti generare celle ad alto rapporto di allungamento, oppure celle degeneri. Si consideri un elemento triangolare di griglia (Figura 34): è possibile dimostrare che l'errore dovuto alla griglia è proporzionale a $(\sin \alpha)^{-1}$, per cui al tendere di α a zero l'errore tende all'infinito;
- un grid non strutturato è in genere più pesante di uno strutturato, essendo generato un numero maggiore di celle,
- i grid strutturati consentono soluzioni più precise.

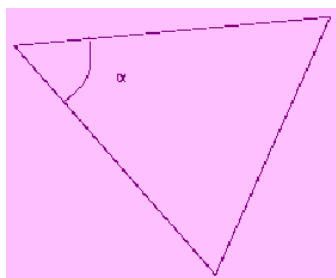


Figura 34

In base alle necessità di calcolo, tempo a disposizione, grado di approssimazione ricercato, l'utente dei programmi di generazione di griglie può decidere la strategia idonea da impiegarsi per un particolare problema, ricordando che una buona soluzione numerica di una PDE è sempre costruita su di un buon grid.