

POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale e dell'Informazione

Corso di Laurea in
Ingegneria Aeronautica



Stanford University Unstructured: un software innovativo per la simulazione e l'ottimizzazione geometrica

Relatore: Prof. Maurizio QUADRIO

Correlatori: Ing. Samuele ZAMPINI

Ing. Fabrizio TESSICINI

Tesi di Laurea di:

Roberto PIERI Matr. 765292

Anno Accademico 2012 – 2013

Indice

Introduzione	1
1 Le equazioni della fluidodinamica	3
1.1 Descrizione matematica del flusso	3
1.2 Sistema completo delle equazioni di Navier-Stokes	9
1.3 Turbolenza	11
1.3.1 Procedure di media	14
1.3.2 Equazioni di Navier-Stokes mediate alla Favre-Reynolds	16
1.3.3 Modelli di turbolenza	20
2 Ottimizzazione geometrica	27
2.1 Metodo aggiunto	29
2.1.1 Ottimizzazione della forza aerodinamica	35
2.1.2 Ipotesi di frozen viscosity	37
2.2 Definizione delle variabili di progetto	39
2.2.1 Funzioni di forma di Hicks-Henne	39
2.2.2 Metodo FFD	41
2.3 Algoritmo di ottimizzazione vincolata	43
2.3.1 Metodo risolutivo	43
2.3.2 I metodi Sequential Quadratic Programming	44
3 Il codice SU^2	47
3.1 Struttura del codice	48
3.2 Formato nativo della mesh	51
3.3 Discretizzazione numerica	54
3.3.1 Integrazione spaziale	54
3.3.2 Integrazione temporale	57
3.4 Metodo Multigrid	58
3.5 Deformazione della mesh	59
3.6 Comandi per l'avvio del software	61

4 Risultati	65
4.1 La mesh	65
4.2 Simulazioni non viscosi	66
4.2.1 Caso 1: Profilo NACA 0012	66
4.2.2 Caso 2: ONERA M6	74
4.3 Simulazioni viscosi	81
4.3.1 Caso 3: NACA 0012	81
4.3.2 Caso 4: NACA 0012	88
Conclusioni	97
Bibliografia	99

Elenco delle figure

1.1	Descrizione del volume di controllo	4
1.2	Dominio della soluzione di una DNS nello spazio dei numeri d'onda	12
2.1	Ciclo di ottimizzazione geometrica.	28
2.2	Confronto tra la sensitività calcolata tramite ipotesi di Frozen Viscosity, differenze finite e calcolo aggiunto completamente turbolento	38
2.3	Confronto tra l'ottimizzazione con l'ipotesi di Frozen Viscosity e caso totalmente turbolento [4].	38
2.4	Funzioni di Hicks-Henne.	40
2.5	Deformazione tramite funzioni di Hicks-Henne.	40
2.6	Parallelepipedo indeformato [18].	41
2.7	Parallelepipedo deformato [18].	42
2.8	Parallelepipedo deformato e punti di controllo [18].	42
3.1	Gerarchia delle classi del modulo SU2_CFD	48
3.2	Classi figlie di CSolution e CVariable	50
3.3	Classi figlie di CNumerics	51
3.4	Struttura di una simulazione RANS con integrazione esplicita nel tempo.	52
3.5	Griglia primaria e duale [14].	55
3.6	Schematizzazione del metodo a molle torsionali [6].	61
4.1	Dominio di calcolo.	66
4.2	Griglia di calcolo intorno al NACA0012.	67
4.3	Soluzione intorno al NACA 0012.	68
4.4	Coefficiente di pressione sul profilo e riferimento [20].	68
4.5	Residuo del calcolo diretto.	69
4.6	Sensitività aerodinamica normalizzata sul profilo	70
4.7	Ingrandimento della sensitività sul profilo.	70
4.8	Andamento del residuo dell'equazione di continuità aggiunta.	71

4.9	Andamento dei coefficienti aerodinamici durante il ciclo di ottimizzazione.	72
4.10	Soluzione intorno al NACA 0012 ottimizzato.	73
4.11	Coefficiente di pressione sul profilo originale e ottimizzato. . .	73
4.12	Deformazione del profilo.	74
4.13	Dominio di calcolo ed elementi di griglia sull'ala.	75
4.14	Soluzione sull'ala.	75
4.15	Coefficiente di pressione sull'ala ottenuto con SU^2 e confrontato con i dati di riferimento [19].	76
4.16	Sensitività geometrica alle variabili di progetto e contour del numero di Mach sull'ala.	77
4.17	Box delle variabili di progetto.	78
4.18	Andamento dei coefficienti aerodinamici durante il ciclo di ottimizzazione.	78
4.19	Confronto del numero Mach, del coefficiente di pressione e della deformazione dell'ala.	79
4.20	Confronto tra il coefficiente di pressione dell'ala indeformata e dell'ala ottimizzata.	80
4.21	Dominio di calcolo.	81
4.22	Griglia di calcolo intorno al NACA0012.	82
4.23	Soluzione intorno al NACA 0012.	83
4.24	Coefficiente di pressione confrontato con il riferimento [5]. . . .	83
4.25	Andamento del residuo dell'equazione di continuità.	84
4.26	Variabile aggiunta Ψ_2	85
4.27	Sensitività aerodinamica sul profilo	85
4.28	Andamento del residuo dell'equazione di continuità aggiunta. .	86
4.29	Deformazione del profilo.	87
4.30	Coefficiente di pressione del profilo originale e deformato. . . .	88
4.31	Dominio di calcolo.	89
4.32	Griglia di calcolo intorno al NACA0012.	90
4.33	Soluzione intorno al profilo.	90
4.34	Coefficiente di pressione confrontato con il caso di riferimento [19].	91
4.35	Sensitività aerodinamica sul profilo	92
4.36	Andamento dei coefficienti aerodinamici nel ciclo di ottimizzazione.	93
4.37	Coefficiente di pressione sul profilo indeformato e profilo ottimizzato.	94
4.38	Soluzione intorno al profilo ottimizzato.	94
4.39	Deformazione del profilo.	95

Elenco delle tabelle

4.1	Scomposizione del coefficiente di resistenza nei due contributi dovuti all'attrito (C_{D_f}) e alla pressione (C_{D_P}).	88
4.2	Scomposizione del coefficiente di resistenza nei due contributi dovuti all'attrito (C_{D_f}) e alla pressione (C_{D_P}).	93

Sommario

Il presente lavoro di tesi si pone l'obiettivo di valutare le capacità del nuovo software libero di simulazione fluidodinamica SU^2 , sviluppato dall'Università di Stanford e attualmente alla sua seconda release ufficiale. La particolarità di questo programma è la presenza dell'ottimizzatore geometrico che permette, attraverso un algoritmo di ottimizzazione *gradient based*, di minimizzare una funzione obiettivo scelta dall'utente. Il software presenta diversi moduli, in linguaggio C++, che svolgono compiti diversi come ad esempio risoluzione delle equazioni, deformazione della mesh o scomposizione del dominio di calcolo. Sono poi presenti programmi scritti in linguaggio Python che combinano le funzioni dei moduli descritti prima per eseguire lavori più complessi, come per l'appunto l'ottimizzazione geometrica. Quest'ultimo si basa su un algoritmo che necessita dei gradienti della funzione obiettivo e degli eventuali vincoli, i quali vengono calcolati mediante le equazioni del metodo aggiunto continuo. Questo metodo per il calcolo dei gradienti prevede la linearizzazione delle equazioni del flusso che, sotto opportune condizioni al contorno, permettono il calcolo dei gradienti ad un costo computazionale trascurabile. Inoltre diverse tecniche per la deformazione della geometria da ottimizzare e di deformazione della mesh sono implementate nel codice. In questo lavoro di tesi si applicheranno le funzioni del codice per eseguire l'ottimizzazione di casi differenti dal punto di vista della modellazione matematica, ma anche della fisica in gioco. A tale proposito verrà applicata la procedura prima a casi non viscosi, prima ad profilo NACA 0012 in regime transonico, poi all'ONERA M6, sempre in regime transonico. Dopo si passerà ai casi viscosi, entrambi bidimensionali (NACA 0012) prima in regime laminare e poi turbolento mediante l'utilizzo del modello di turbolenza di Spalart-Allmaras implementato nel codice. Si verificherà che la procedura di ottimizzazione possa essere portata a buon fine in maniera autonoma dal programma.

Parole chiave: Ottimizzazione geometrica, metodo aggiunto.

Abstract

The present work aims to evaluate the capabilities of the new CFD software *SU²*, developed by Stanford University and currently at its second official release. The peculiarity of this software is the presence of shape optimization tool which allows minimization of an objective function chosen by the user, using a gradient-based optimization algorithm. The software is structured in different modules, written in C++ programming language, which perform different tasks such as solving equations, mesh deformation or domain decomposition. There are also Python programs combining the functions of modules described above to perform more complex tasks, such as shape optimization. The letter is based on an algorithm that requires gradient evaluations of objective functions and constraints which are computed by the continuous adjoint method. This way to compute gradients provides for the linearization of flow equations that allows the calculation of the gradients to a negligible computational cost. Furthermore, different techniques for the geometry and mesh deformation have been implemented in code.

In this work functions to perform shape optimization will be applied on different cases not only from the point of view of mathematical modeling, but also from the physical point of view. In this regard a first application of the procedure will be applied to inviscid cases, prior to NACA 0012 airfoil in transonic regime, then on ONERA M6 in transonic regime. The second step is to apply the procedure to viscous bidimensional cases (NACA 0012 profile), the first in laminar regime, then in turbulent regime through the use of turbulence model of Spalart-Allmaras implemented in the code. It will be verified that the optimization procedure can be seen out independently by the program.

Key words: Shape optimization, adjoint method

Introduzione

Il classico approccio al problema dell'ottimizzazione, in particolare dell'ottimizzazione geometrica nel campo industriale, è quello di procedere per tentativi basandosi principalmente sul *know-how* dell'azienda; questo rappresenta il tipico modo di innovazione incrementale che porta ad un miglioramento spesso marginale ad ogni passo del processo innovativo, ma riduce al minimo i rischi di fallimento. Un'idea ambiziosa è quella di automatizzare questo processo senza però perderne le caratteristiche principali; basandosi però sulle conoscenze tecnico-scientifiche non è possibile rendere il processo automatico. L'approccio aggiunto rappresenta da questo punto di vista uno strumento assolutamente adeguato allo scopo; esso infatti rappresenta la base degli algoritmi di ottimizzazione cosiddetti *gradient-based*, cioè algoritmi che necessitano delle derivate della funzione da ottimizzare per valutare la deformazione necessaria a generare la variazione maggiore della funzione obiettivo.

In quest'ottica nasce il presente lavoro di tesi in collaborazione con la Scuderia Toro Rosso di Formula 1, interessata ad uno strumento che possa automatizzare il processo di ottimizzazione geometrica. Come noto, infatti, in Formula 1 ogni anno vengono apportate modifiche al regolamento che, seppur minime, sono tali da variare gli equilibri sulla monoposto. Inoltre il progetto iniziale dell'autovettura, durante il corso di tutto il campionato, viene modificato costantemente in base al tipo di tracciato, allo stile di guida del pilota e, soprattutto negli ultimi anni, al consumo gomme; avere quindi uno strumento che possa indicare le zone più sensibili, tra quelle ammesse, sulle quali poter agire e magari anche avere a disposizione un algoritmo automatico che imposti il ciclo di ottimizzazione rappresenta un grande vantaggio.

A tale proposito il software SU^2 sviluppato presso l'Università di Stanford e recentemente apparso sulla scena rappresenta, nel panorama dei codici CFD open-source, l'unico esempio di codice di calcolo che nasce con un *toolset* di strumenti di ottimizzazione di forma basati sull'operatore aggiunto. Questo lavoro di tesi rappresenta forse il primo lavoro eseguito all'esterno del gruppo degli sviluppatori, e si pone l'obiettivo di valutare le potenzialità di SU^2 , in particolare la capacità di condurre l'ottimizzazione in maniera autonoma e

con successo. La tesi descrive alcuni casi, di complessità crescente e caratterizzati da una distanza crescente dal problema-tipo per cui SU^2 è noto, ovvero calcoli non viscosi su geometrie di interesse aeronautico in regime transonico.

Di seguito viene proposto l'inquadramento matematico delle equazioni del flusso nel Capitolo 1, con particolare attenzione alle equazioni turbolente e ai modelli di turbolenza. Nel Capitolo 2 si introduce il concetto delle equazioni aggiunte e si espongono tutti i passi del processo di ottimizzazione; inoltre saranno introdotte le tecniche di deformazione della mesh utilizzate per le simulazioni e le variabili di progetto necessarie per la deformazione della superficie. Nel Capitolo 3 viene introdotta la struttura del codice utilizzato e la numerica implementata nelle diverse classi del programma. Infine nell'ultimo Capitolo si propongono i risultati numerici dei differenti casi analizzati, ponendo maggiore attenzione al processo di ottimizzazione geometrica.

Capitolo 1

Le equazioni della fluidodinamica

Vengono di seguito introdotte le equazioni di Navier-Stokes complete per un fluido newtoniano. Verrà poi esposta la formulazione delle equazioni mediate alla Favre-Reynolds, focalizzando l'attenzione sulla chiusura di queste ultime mediante i modelli di turbolenza. Dopo una breve introduzione sulle varie tipologie di modelli verrà presentato con maggiore dettaglio il modello Spalart-Allmaras, l'unico ad ora disponibile in SU^2 .

1.1 Descrizione matematica del flusso

La *fluidodinamica* è lo studio del movimento di un numero di atomi o molecole; si suppone quindi che la densità del fluido sia elevata abbastanza da poterlo considerare un continuo. Assunta valida l'ipotesi di continuo, le equazioni che regolano la dinamica dei fluidi derivano direttamente dalle seguenti leggi di conservazione:

1. conservazione della massa,
2. conservazione della quantità di moto,
3. conservazione dell'energia.

La conservazione di una qualsiasi quantità del fluido implica che la variazione totale della stessa, all'interno di un volume di controllo arbitrario, sia pari alla somma del flusso netto della quantità attraverso il bordo del volume, delle forze interne e delle eventuali sorgenti e forze esterne agenti sul volume di controllo.

Tale volume di controllo Ω è definito come la regione arbitraria di fluido delimitata dalla superficie chiusa $\partial\Omega$ e fissa nello spazio (punto di vista *euleriano*). È possibile definire anche la superficie infinitesima dS e il versore associato normale \mathbf{n} , come mostrato nella Figura seguente.

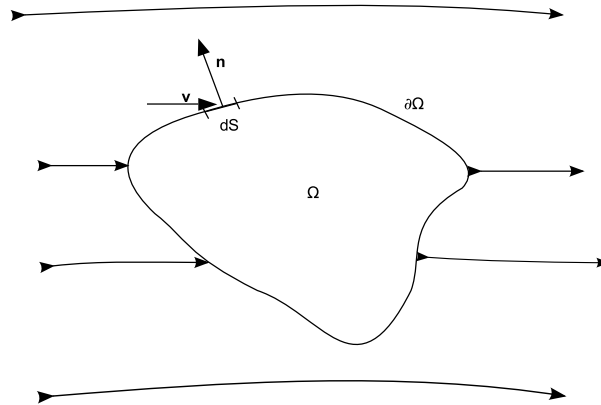


Figura 1.1: Descrizione del volume di controllo

Di seguito si introdurranno i passaggi matematici per giungere alla forma integrale delle leggi di conservazione sopra elencate [3].

Conservazione della massa

La legge di conservazione della massa stabilisce che non si può generare o distruggere massa all'interno del sistema fluido.

Si consideri quindi il modello di volume di controllo definito in Figura 1.1. In questo caso la quantità conservata è la densità ρ . La variazione temporale della massa all'interno del volume di controllo si può esprimere come:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \, d\Omega.$$

Per la determinazione del flusso netto di massa che attraversa il volume si consideri un elemento infinitesimo di superficie dS , e sia \mathbf{v} il vettore velocità del fluido e \mathbf{n} il versore normale alla superficie; il flusso di massa che attraversa la superficie dS sarà:

$$-\rho(\mathbf{v} \cdot \mathbf{n})dS,$$

dove per convenzione il versore normale è diretto verso l'esterno del volume di controllo, quindi il flusso è entrante quando il prodotto scalare $\mathbf{v} \cdot \mathbf{n}$ è negativo.

Questi sono gli unici termini presenti nell'equazione di continuità poiché non sono presenti sorgenti per definizione; è quindi possibile scrivere:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \, d\Omega + \oint_{\partial\Omega} \rho(\mathbf{v} \cdot \mathbf{n}) \, dS = 0, \quad (1.1)$$

che rappresenta la forma integrale della legge di conservazione della massa, o equazione di continuità.

Equazione della quantità di moto

È possibile ricavare l'equazione partendo dal secondo principio della dinamica: la variazione di quantità di moto è causata dalle forze agenti sull'elemento di massa. Sempre facendo riferimento alla definizione di volume di controllo (Figura 1.1), la quantità di moto di un elemento infinitesimo del volume sarà:

$$\rho \mathbf{v} d\Omega,$$

la cui variazione nel tempo in tutto il volume è:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{v} d\Omega.$$

La quantità conservata è la quantità di moto per unità di volume $\rho \mathbf{v}$ e il flusso può essere espresso come:

$$- \oint_{\partial\Omega} \rho \mathbf{v} (\mathbf{v} \cdot \mathbf{n}) dS,$$

con la solita convenzione sulla normale e sul segno del flusso entrante.

Le forze agenti sul fluido sono di due tipi:

- Forze di volume: sono forze che agiscono direttamente sulla massa, come la forza di gravità, di Coriolis o la forza centrifuga e sono definite come forze per unità di volume: $\rho \mathbf{f}_e$;
- Forze di superficie: agiscono sul bordo del volume e derivano dalla distribuzione di pressione o dagli sforzi dovuti all'attrito e sono forze per unità di superficie.

Il contributo delle forze di volume è:

$$\int_{\Omega} \rho \mathbf{f}_e d\Omega.$$

Le forze di superficie sono dette sforzi e possono essere divise in due parti, una componente isotropa di pressione e una componente di sforzi viscosi:

$$\underline{\underline{F}}_S = -p\underline{\underline{I}} + \underline{\underline{\sigma}}, \quad (1.2)$$

dove $\underline{\underline{I}}$ è il tensore identità e $\underline{\underline{\sigma}}$ il tensore degli sforzi viscosi, che sarà trattato successivamente con maggiore attenzione.

È ora possibile mettere insieme tutti gli elementi per scrivere l'equazione integrale della conservazione della quantità di moto:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{v} d\Omega + \oint_{\partial\Omega} \rho \mathbf{v} (\mathbf{v} \cdot \mathbf{n}) dS = - \oint_{\partial\Omega} p \mathbf{n} dS + \oint_{\partial\Omega} (\underline{\underline{\sigma}} \cdot \mathbf{n}) dS. \quad (1.3)$$

Equazione dell'energia

L'equazione dell'energia si ricava dal primo principio della termodinamica, il quale stabilisce che la variazione nel tempo dell'energia totale all'interno del volume di controllo Ω è dovuta al lavoro delle forze agenti sul volume nell'unità di tempo e al flusso netto di calore all'interno di esso.

L'energia totale per unità di massa è definita come la somma dell'energia interna specifica (e) e dell'energia cinetica specifica:

$$E = e + \frac{|\mathbf{v}|^2}{2}. \quad (1.4)$$

La quantità conservata è l'energia totale E , quindi la sua variazione nel volume si può scrivere come:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega.$$

Il flusso attraverso il bordo di Ω è:

$$- \oint_{\partial\Omega} \rho E (\mathbf{v} \cdot \mathbf{n}) dS,$$

con la solita convenzione per la direzione del versore normale.

A differenza delle due equazioni precedenti, nell'equazione dell'energia è presente anche un flusso diffusivo, proporzionale al gradiente della quantità conservata; siccome il flusso diffusivo è definito solo per fluidi a riposo, sarà proporzionale alla sola energia interna:

$$-\gamma \rho \kappa \nabla e$$

dove $\gamma = \frac{c_p}{c_v}$ è il rapporto tra i coefficienti di calore specifico e κ è il coefficiente di diffusività termica. Questo termine, che rappresenta il flusso di calore dovuto al gradiente termico, può essere espresso più semplicemente nella forma della legge di Fourier:

$$-k \nabla T$$

con k coefficiente di conducibilità termica.

Il termine di sorgente di volume è composto dal lavoro per unità di tempo delle forze di volume e dal flusso di calore nell'unità di tempo dovuto all'irraggiamento:

$$Q_V = \rho \mathbf{f}_e \cdot \mathbf{v} + \dot{q}_h.$$

Rimane ora solo il contributo delle forze di superficie, che interviene come lavoro nell'unità di tempo delle stesse e può essere espresso come:

$$\mathbf{Q}_S = -p\mathbf{v} + \underline{\underline{\sigma}} \cdot \mathbf{v}. \quad (1.5)$$

Unendo tutti i termini precedenti si giunge all'equazione dell'energia:

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega + \oint_{\partial\Omega} \rho E (\mathbf{v} \cdot \mathbf{n}) dS &= \oint_{\partial\Omega} k (\nabla T \cdot \mathbf{n}) dS + \oint_{\partial\Omega} (\rho \mathbf{f}_e \cdot \mathbf{v} + \dot{q}_h) dS \\ &\quad - \oint_{\partial\Omega} p (\mathbf{v} \cdot \mathbf{n}) dS + \oint_{\partial\Omega} (\underline{\underline{\sigma}} \cdot \mathbf{v}) \cdot \mathbf{n} dS. \end{aligned} \quad (1.6)$$

Solitamente l'equazione (1.6) viene proposta in una forma leggermente differente. A questo proposito si introduce l'entalpia totale H e la relazione che lega questa variabile termodinamica all'energia totale e alla pressione:

$$H = h + \frac{|\mathbf{v}|^2}{2} = E + \frac{p}{\rho}. \quad (1.7)$$

Raccogliendo insieme il flusso di energia e il termine di pressione, utilizzando la (1.7) si può giungere alla forma più comune dell'equazione di conservazione dell'energia:

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega + \oint_{\partial\Omega} \rho H (\mathbf{v} \cdot \mathbf{n}) dS &= \oint_{\partial\Omega} k (\nabla T \cdot \mathbf{n}) dS \\ &\quad + \int_{\Omega} (\rho \mathbf{f}_e \cdot \mathbf{v} + \dot{q}_h) d\Omega + \oint_{\partial\Omega} (\underline{\underline{\sigma}} \cdot \mathbf{v}) \cdot \mathbf{n} dS. \end{aligned} \quad (1.8)$$

Si è quindi giunti alla formulazione completa delle equazioni di governo del moto di un fluido.

Prima di procedere è necessaria una riflessione su tali equazioni; infatti la forma integrale, detta anche *debole*, delle leggi di conservazione ha due proprietà fondamentali:

- se non sono presenti sorgenti di volume, la variazione delle quantità conservate dipende esclusivamente dai flussi sul bordo $\partial\Omega$ e non dai flussi interni al volume;
- le equazioni sono valide anche in presenza di discontinuità nel flusso, ad esempio onde d'urto o discontinuità di contatto.

È quindi evidente che, vista la sua generalità e le proprietà sopra descritte, questa sia la forma più semplice da implementare in un codice fluidodinamico.

Tensore degli sforzi viscosi

Nei paragrafi precedenti è stato introdotto il tensore degli sforzi viscosi, vediamo ora come esso viene trattato dal punto di vista matematico.

Quando viene isolato un elemento di fluido bisogna tener conto delle forze di superficie generate dal fluido circostante; le forze per unità di superficie sono dette sforzi.

Gli sforzi si scompongono in una parte di pressione isotropa e una parte *deviatorica* dovuta all'attrito. Il tensore degli sforzi viscosi, considerando un sistema di coordinate cartesiane ortogonali xyz , è il seguente:

$$\underline{\underline{\sigma}} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}, \quad (1.9)$$

dove la notazione σ_{ij} per convenzione stabilisce che la componente di sforzo agisce su un piano ortogonale all'asse i in direzione dell'asse j . Quindi i termini diagonali rappresentano sforzi normali, quelli fuori diagonale rappresentano sforzi di taglio.

L'espressione di tali componenti di sforzo dipende dalle proprietà dinamiche del fluido; in particolare, fluidi come aria e acqua, i cui sforzi sono proporzionali al gradiente di velocità, sono detti *fluidi Newtoniani* e le componenti di sforzo si possono esprimere come:

$$\sigma_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \lambda \frac{\partial v_k}{\partial x_k} \delta_{ij} \quad (1.10)$$

dove δ_{ij} è il δ di Kronecker, λ rappresenta il secondo coefficiente di viscosità e μ la viscosità dinamica. Introducendo il coefficiente di viscosità di volume μ_v e l'ipotesi di Stokes ($\mu_v = 0$):

$$\mu_v = \lambda + \frac{2}{3}\mu = 0, \quad (1.11)$$

si giunge ad una relazione tra μ e λ : $\lambda = -\frac{2}{3}\mu$. Bisogna sottolineare che non ci sono prove sperimentali che neghino l'ipotesi di Stokes se non ad elevatissime pressioni o temperature, per cui la relazione (1.11) viene utilizzata per eliminare il secondo coefficiente di viscosità dalle equazioni:

$$\sigma_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{2}{3}\mu \frac{\partial v_k}{\partial x_k} \delta_{ij} \quad (1.12)$$

1.2 Sistema completo delle equazioni di Navier-Stokes

Nella sezione precedente sono state ricavate le equazioni di bilancio che governano la dinamica dei fluidi, è ora possibile riscriverle tutte in un unico sistema di equazioni, in una forma più generale e con l'aggiunta delle relazioni termodinamiche per la chiusura delle equazioni. Inoltre il sistema sarà proposto sia in forma integrale che in forma differenziale.

A tale proposito si introducono il tensore dei flussi convettivi \mathbf{F}^c , i tensori dei flussi viscosi divisi in contributo delle forze viscosi \mathbf{F}^{v1} e del trasferimento di calore \mathbf{F}^{v2} , sia inoltre \mathbf{Q} il vettore dei termini sorgenti; allora il sistema completo delle equazioni di conservazione in forma integrale è:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega + \oint_{\partial\Omega} [\mathbf{F}^c - (\mathbf{F}^{v1} + \mathbf{F}^{v2})] \cdot \mathbf{n} dS = \int_{\Omega} \mathbf{Q} d\Omega, \quad (1.13)$$

dove \mathbf{U} è il vettore delle *variabili conservative*:

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{Bmatrix}, \quad (1.14)$$

mentre i flussi sono definiti come:

$$\mathbf{F}^c = \begin{Bmatrix} \rho \mathbf{v}^T \\ \rho \mathbf{v} \otimes \mathbf{v} + p \underline{I} \\ \rho H \mathbf{v}^T \end{Bmatrix} \quad (1.15)$$

e

$$\mathbf{F}^{v1} = \begin{Bmatrix} \underline{\underline{0}} \\ \underline{\underline{\sigma}} \\ (\underline{\underline{\sigma}} \cdot \mathbf{v})^T \end{Bmatrix} \quad \mathbf{F}^{v2} = \begin{Bmatrix} \underline{\underline{0}} \\ \underline{\underline{0}} \\ (k \nabla T)^T \end{Bmatrix}. \quad (1.16)$$

Infine i termini sorgente sono:

$$\mathbf{Q} = \begin{Bmatrix} 0 \\ \rho \mathbf{f}_e \\ \rho \mathbf{f}_e \cdot \mathbf{v} + \dot{q}_h \end{Bmatrix}. \quad (1.17)$$

L'equazione rappresenta un sistema di cinque equazioni nelle cinque variabili conservative, ma sono presenti sette incognite: ρ , \mathbf{v} , E , p , T . È quindi necessario aggiungere altre due equazioni, che dovranno essere relazioni termodinamiche tra le variabili di stato, per chiudere il problema e inoltre fornire una relazione per la viscosità dinamica μ e una per la conducibilità termica k .

Si ricorre quindi alla formulazione per i gas perfetti poiché, in aerodinamica classica, è ragionevole ritenere di lavorare con fluidi che si comportano come se fossero gas caloricamente perfetti. Per tali gas la legge di stato è la seguente:

$$p = \rho RT \quad (1.18)$$

con R la costante specifica del gas. L'entalpia e l'energia interna specifiche sono definite come:

$$\begin{aligned} h &= c_p T \\ e &= c_v T. \end{aligned} \quad (1.19)$$

Combinando l'equazione (1.7) dell'entalpia totale con (1.18) e (1.19) è possibile scrivere la pressione come:

$$p = (\gamma - 1) \rho \left(E - \frac{|\mathbf{v}|^2}{2} \right), \quad (1.20)$$

mentre la temperatura si ricava dall'equazione di stato.

Per il coefficiente di viscosità si utilizza la legge di Sutherland:

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T + S}{T_0 + S}, \quad (1.21)$$

dove $\mu_0 = 1.716 \times 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}$, $T_0 = 273.15 \text{ K}$ e $S = 110.4 \text{ K}$. Mentre per quanto riguarda il coefficiente di conducibilità termica si ritiene che sia costante in tutto il fluido, tale ipotesi è sicuramente vera per i liquidi, ma è anche usata per l'aria, e vale:

$$k = c_p \frac{\mu}{\text{Pr}} \quad (1.22)$$

dove Pr rappresenta il numero di Prandtl, che si considera anch'esso costante nel fluido e per l'aria vale $\text{Pr} = 0.72$. Si ridefiniscono ora le due componenti dei flussi viscosi raccogliendo la viscosità:

$$\mathbf{F}^{v1} = \left\{ \begin{array}{c} \underline{\underline{0}} \\ \underline{\underline{\tau}} \\ (\underline{\underline{\tau}} \cdot \mathbf{v})^T \end{array} \right\} \quad \mathbf{F}^{v2} = \left\{ \begin{array}{c} \underline{\underline{0}} \\ \underline{\underline{0}} \\ (c_p \nabla T)^T \end{array} \right\}. \quad (1.23)$$

dove τ rappresenta i gradienti di velocità del tensore degli sforzi, ovvero:

$$\underline{\underline{\sigma}} = \mu \underline{\underline{T}}.$$

Definendo:

$$\mu^1 = \mu \text{ e } \mu^2 = \frac{\mu}{\text{Pr}},$$

il sistema delle equazioni di governo in forma integrale si riscrive come:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega + \oint_{\partial\Omega} [\mathbf{F}^c - (\mu^1 \mathbf{F}^{v1} + \mu^2 \mathbf{F}^{v2})] \cdot \mathbf{n} dS = \int_{\Omega} \mathbf{Q} d\Omega. \quad (1.24)$$

Ora il bilancio equazioni-incognite risulta soddisfatto a patto di introdurre le corrette condizioni al contorno e iniziali.

Applicando il teorema di Gauss è possibile riscrivere l'integrale di superficie (il secondo integrale dell'equazione (1.24)) in integrale di volume e l'equazione (1.24) si può riscrivere in forma differenziale:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_c - \nabla \cdot (\mu^1 \mathbf{F}^{v1} + \mu^2 \mathbf{F}^{v2}) = \mathbf{Q}. \quad (1.25)$$

Notazione tensoriale

Per semplicità successiva si riportano le equazioni di governo complete in notazione tensoriale:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho v_j)}{\partial x_j} = 0 \\ \frac{\partial (\rho v_i)}{\partial t} + \frac{\partial (\rho v_j v_i)}{\partial x_j} = -\frac{\partial p}{\partial x_j} \delta_{ij} + \frac{\partial (\mu^1 \tau_{ij})}{\partial x_j} + \rho f_{e_i} \\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho H v_j)}{\partial x_j} = \frac{\partial (\mu^1 \tau_{ij} v_i)}{\partial x_j} + \frac{\partial}{\partial x_j} \left(\mu^2 c_p \frac{\partial T}{\partial x_j} \right) + \rho f_{e_j} v_j + \dot{q}_h \end{cases} \quad (1.26)$$

1.3 Turbolenza

Le equazioni precedenti descrivono completamente il moto del flusso per qualsiasi numero di Reynolds; nel caso di flussi turbolenti però una teoria analitica semplice non esiste, la speranza è quindi di risolvere le equazioni sfruttando la potenza di calcolo dei moderni computer per il calcolo di quantità interessanti dei flussi turbolenti [16]. L'affermazione precedente risulta più chiara riflettendo su alcune proprietà dei flussi turbolenti.

In un flusso turbolento, anche con condizioni al contorno costanti, le grandezze scalari e vettoriali che caratterizzano il flusso non sono stazionarie e

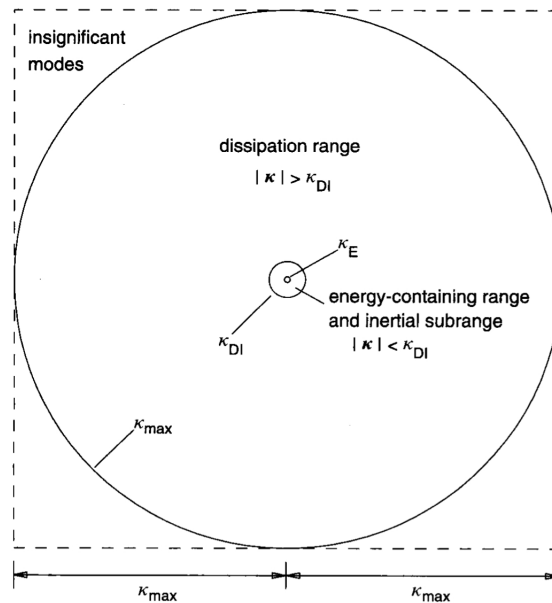


Figura 1.2: Dominio della soluzione di una DNS nello spazio dei numeri d'onda ($k = \frac{2\pi}{l}$); a $Re_\lambda = 70$ solo lo 0.02% dei modi rappresentati è al di sopra del range dissipativo [16].

presentano oscillazioni attorno ad un valor medio. Inoltre l'analisi di un flusso turbolento è complicata poiché occorre seguire l'evoluzione di fenomeni su scale di tempo e lunghezza molto differenti tra loro.

La scelta più comune per l'analisi numerica di un flusso turbolento è quindi la risoluzione delle equazioni tramite calcolatore; a questo scopo esistono tre differenti approcci: DNS, LES, RANS.

La DNS (*Direct Numerical Simulation*) consiste nella risoluzione delle equazioni di Navier-Stokes con una discretizzazione spaziale e temporale tale da risolvere tutte le scale del moto. Tale approccio è concettualmente quello più semplice e fornisce un livello di dettaglio massimo, non vi sono sostanziali perdite di informazione. Bisogna però notare che il costo computazionale è estremamente elevato (cresce come Re^3), motivo per il quale, fino agli anni '70, tale approccio era inapplicabile a causa della scarsa potenza di calcolo a disposizione.

Facendo riferimento alla Figura 1.2, è possibile notare come solo una ridottissima percentuale dei modi risolti con una DNS sia rappresentativa delle scale energeticamente più significative. È quindi evidente che il costo è eccessivamente oneroso quando lo scopo è quello di studiare caratteristiche macroscopiche del flusso.

Un approccio alternativo è dato dalla LES (*Large Eddy Simulation*) che prevede la rappresentazione, senza approssimazioni, delle scale più grandi della turbolenza, mentre le scale minori sono rappresentate tramite appositi modelli numerici. In pratica le LES risolvono le equazioni del moto per un campo di velocità filtrato rappresentativo delle scale maggiori della turbolenza. Dal punto di vista del costo computazionale, le LES sono una via di mezzo tra la DNS e le RANS; tenendo presente che il costo dipende dalle scale che si vogliono rappresentare, idealmente, a parità di costo computazionale di una DNS, si riesce ad ottenere lo stesso livello di dettaglio, cosa praticamente impossibile per le RANS vista la differenza di approccio al problema, come sarà esposto in seguito.

Come discusso precedentemente, la maggior parte del costo computazionale di una DNS è richiesto per la rappresentazione delle scale più piccole (o numeri d'onda grandi se si fa riferimento allo spazio di Fourier), mentre il maggior contenuto energetico è racchiuso nelle scale più grandi del moto; nella LES la dinamica delle scale maggiori è rappresentata esplicitamente, l'influenza delle scale minori è rappresentata da semplici modelli, si evita quindi il costo enorme della rappresentazione delle scale minori della turbolenza.

In molte applicazioni ingegneristiche non è necessario il livello di dettaglio di DNS o LES, molto più semplicemente, interessano proprietà macroscopiche, in generale infatti basta la conoscenza dei valori medi delle quantità fluidodinamiche coinvolte. È quindi conveniente in questi casi utilizzare la formulazione RANS (*Reynolds Averaged Navier-Stokes equations*) che prevede la risoluzione delle equazioni mediate. In particolare si effettua dapprima la scomposizione alla Reynolds delle variabili fluidodinamiche in una parte media ed una fluttuante, poi si procede con la mediazione delle equazioni di Navier-Stokes. L'operazione di media delle equazioni produce un termine aggiuntivo detto *tensore degli sforzi di Reynolds*, che diventa un'incognita delle equazioni mediate, per cui il sistema risultante è sotto-determinato. Per pareggiare il conto equazioni-incognite è necessaria l'introduzione dei cosiddetti modelli di turbolenza che si dividono in modelli basati sull'ipotesi di viscosità turbolenta e altri sulla modellazione diretta del tensore degli sforzi di Reynolds.

I vantaggi di questo approccio sono sicuramente il ridotto costo della simulazione, dovuto al fatto che si sta praticamente cancellando tutta la parte fluttuante del moto del fluido; considerare solo i valori medi ha però dei contro. Infatti gli svantaggi principali delle RANS sono dati dal fatto che è limitante considerare solo il campo medio delle variabili fluidodinamiche (si pensi ad esempio alle strutture coerenti che con questo approccio sono del tutto cancellate), inoltre tutti i modelli di turbolenza sono caratterizzati da

un certo empirismo, il quale impedisce la determinazione a priori di modelli che forniscano previsioni ragionevoli per ogni geometria e condizione di flusso.

Di seguito si riportano le procedure di scomposizione delle variabili, di media delle equazioni e la classificazione delle diverse classi di modelli di turbolenza.

1.3.1 Procedure di media

Il primo passo da eseguire è la scomposizione delle variabili fluidodinamiche in una parte media ed una fluttuante, successivamente si esegue la media delle equazioni di governo. Per flussi a densità variabile esistono due differenti tecniche per la media: la forma non pesata, ovvero la media secondo Reynolds o la forma pesata sulla densità, ovvero media secondo Favre [9, 1], che è più comoda dal punto di vista dell'implementazione nei codici di CFD.

Media temporale secondo Reynolds

Sia $Q(\mathbf{x}, t)$ una variabile continua e definita in $\Omega \subset \mathbb{R}^d$, con $d = 1, 2, 3$; la sua media temporale è data da:

$$\bar{Q}(\mathbf{x}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T Q(\mathbf{x}, t) dt.$$

La decomposizione alla Reynolds è:

$$Q(\mathbf{x}, t) = \bar{Q}(\mathbf{x}) + Q'(\mathbf{x}, t), \quad (1.27)$$

dalla quale risulta che $Q'(\mathbf{x}, t)$ è la parte fluttuante e $\bar{Q}(\mathbf{x})$ è la media temporale, quindi indipendente dal tempo; di conseguenza la parte fluttuante racchiude tutta l'instazionarietà del fenomeno. Per la definizione di media temporale appena introdotta: $\bar{Q}'(\mathbf{x}, t) = 0$ e $\bar{\bar{Q}}(\mathbf{x}) = \bar{Q}(\mathbf{x})$.

Per le operazioni successive si introduce la media temporale del prodotto di due variabili ϕ e ψ che è la seguente:

$$\overline{\phi\psi} = \overline{(\bar{\phi} + \phi')(\bar{\psi} + \psi')} = \bar{\phi}\bar{\psi} + \bar{\phi}\bar{\psi}' + \bar{\phi}'\bar{\psi} + \bar{\phi}'\bar{\psi}' = \bar{\phi}\bar{\psi} + \bar{\phi}'\bar{\psi}'. \quad (1.28)$$

Non essendoci a priori nessuna ragione per la quale la media temporale di due variabili fluttuanti debba essere nulla, si può affermare che la media temporale del prodotto di due quantità non è semplicemente data dal prodotto delle medie, ma anche da un termine aggiuntivo del tipo $\bar{\phi}'\bar{\psi}'$.

Media temporale secondo Favre

La media alla Reynolds è utilizzata comunemente per flussi a densità costante, invece quando si trattano flussi comprimibili, la media alla Favre diventa la scelta migliore poiché semplifica notevolmente la forma delle equazioni mediate.

Facendo sempre riferimento ad una generica variabile $Q(\mathbf{x}, t)$, e introducendo anche il campo di densità $\rho(\mathbf{x}, t)$, la quantità $\tilde{Q}(\mathbf{x})$, mediata nel tempo e pesata sulla densità, è definita come:

$$\tilde{Q}(\mathbf{x}) = \frac{1}{\bar{\rho}} \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \rho(\mathbf{x}, t) Q(\mathbf{x}, t) dt.$$

ovvero

$$\bar{\rho} \tilde{Q} = \overline{\rho Q} \quad (1.29)$$

dove $\bar{\rho}$ è la densità mediata alla Reynolds.

Quindi la media alla Favre consiste in una scomposizione della grandezza considerata in una parte media pesata sulla densità, $\tilde{Q}(\mathbf{x})$, non più funzione del tempo, e in una parte fluttuante, $Q''(\mathbf{x}, t)$:

$$Q(\mathbf{x}, t) = \tilde{Q}(\mathbf{x}) + Q''(\mathbf{x}, t) \quad (1.30)$$

bisogna sottolineare che, a differenza della scomposizione alla Reynolds, il termine $\overline{Q''}$ è in generale non nullo.

Scomponendo il termine di destra della (1.29) secondo la decomposizione alla Favre (1.30) si ottiene:

$$\overline{\rho Q} = \overline{\rho (\tilde{Q} + Q'')} = \overline{\rho \tilde{Q}} + \overline{\rho Q''} = \bar{\rho} \tilde{Q} + \overline{\rho Q''}.$$

A priori non è possibile annullare il secondo termine del membro di destra, ma, sfruttando la relazione (1.29), $\overline{\rho Q''}$ si deve annullare per verificare l'uguaglianza. Tale proprietà si rivelerà utile nel processo di media delle equazioni di governo.

Come si noterà dai passaggi successivi, la media secondo Favre elimina le fluttuazioni della densità dalle equazioni mediate, rendendo più semplice l'implementazione delle stesse. Per chiarire meglio tale affermazione si consideri il termine convettivo delle equazioni di Navier-Stokes ρuv e si utilizzi rispettivamente la media alla Reynolds e alla Favre:

$$\begin{aligned} \overline{\rho uv} &= \bar{\rho} \bar{u} \bar{v} + \overline{\rho' u' v'} + \overline{u \rho' v'} + \overline{v \rho' u'} + \overline{\rho' u' v'} \\ \overline{\rho uv} &= \bar{\rho} \tilde{u} \tilde{v} + \overline{\rho u'' v''} = \bar{\rho} \tilde{u} \tilde{v} + \overline{\rho u'' v''} \end{aligned}$$

Confrontando le equazioni precedenti si può notare come entrambe presentino al membro di destra le quantità mediate più altri termini legati alle

fluttuazioni. Bisogna però notare che i tre termini addizionali, che coinvolgono le fluttuazioni di densità nell'equazione mediata secondo Reynolds non sono presenti nella seconda equazione, mediata alla Favre, che invece incorpora l'influenza di questi differenti modi di scambio di quantità di moto in meno termini, con maggiore semplicità di scrittura delle equazioni e chiarezza fisica. Infatti si può notare che la forma della seconda equazione è simile a quella di un flusso a densità costante, per il quale vale: $\overline{uv} = \bar{u}\bar{v} + \overline{u'v'}$.

1.3.2 Equazioni di Navier-Stokes mediate alla Favre-Reynolds

Vengono ora riportati i procedimenti per ricavare le equazioni mediate utilizzando le procedure di media descritte precedentemente. Per ottenere tali equazioni sarà necessario scomporre in maniera differente le variabili del sistema, in particolare per ρ e p si utilizzerà la scomposizione secondo Reynolds, mentre per \mathbf{v} , T , E ed H la media alla Favre.

Equazione di continuità mediata

Per dimostrare la semplificazione nella scrittura delle equazioni ottenuta tramite la decomposizione mediata sulla densità, verranno proposte, solo nel caso dell'equazione di continuità, entrambe le scomposizioni. Si consideri quindi la prima equazione del sistema (1.26) e si scompongano in parte media e fluttuante alla Reynolds le variabili ρ e v_j , come descritto precedentemente, e si medino tutti i termini dell'equazione:

$$\frac{\partial}{\partial t} \overline{(\bar{\rho} + \rho')} + \frac{\partial}{\partial x_j} \overline{(\bar{\rho} + \rho') (\bar{v}_j + v'_j)} = 0.$$

Applicando le proprietà di media ricavate precedentemente si ottiene:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \bar{v}_j + \overline{\rho' v'_j}) = 0.$$

Si noti come il risultato dell'equazione precedente non è equivalente ad applicare l'equazione di governo ai valori medi $\bar{\rho}$ e \bar{v}_j ; infatti nell'equazione è presente un termine aggiuntivo all'interno dell'operatore di divergenza. Sarà quindi necessaria, al fine di chiudere le equazioni, un'approssimazione o modellazione di tale termine, aggiungendo ulteriore empirismo all'approccio RANS.

Si riporta ora la scomposizione secondo Favre mostrando quanto essa semplifichi di molto l'equazione. Si eseguono quindi gli stessi procedimenti visti

in precedenza con la differenza che la variabile v_j viene scomposta secondo Favre, si ottiene:

$$\frac{\partial}{\partial t}(\overline{\bar{\rho} + \rho'}) + \frac{\partial}{\partial x_j} \left[\overline{\rho (\tilde{v}_j + v_j'')} \right] = 0$$

dalla quale, sfruttando le proprietà di media mostrate precedentemente, si arriva facilmente all'equazione mediata voluta:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} (\overline{\bar{\rho} \tilde{v}_j}) = 0. \quad (1.31)$$

Si noti come ora il termine aggiuntivo all'interno dell'operatore divergenza non sia più presente e l'espressione è formalmente identica all'equazione di continuità non mediata.

Equazione della quantità di moto mediata

Nelle prossime due equazioni si vedrà l'introduzione di termini incogniti aggiuntivi che bisognerà trattare separatamente.

Si consideri l'equazione della quantità di moto del sistema (1.26) e, come prima, si applichino le due differenti scomposizioni delle variabili con successiva media dell'equazione; per semplificare la trattazione si considerino trascurabili i termini legati alle forze esterne:

$$\frac{\partial}{\partial t} (\overline{\rho (\tilde{v}_i + v_i'')}) + \frac{\partial}{\partial x_j} (\overline{\rho (\tilde{v}_j + v_j'') (\tilde{v}_i + v_i'')}) = - \frac{\partial}{\partial x_i} (\overline{\bar{p} + p''}) + \frac{\partial (\overline{\bar{\mu} \tau_{ij}})}{\partial x_j}$$

Il primo termine del membro di sinistra può essere scomposto come segue:

$$\frac{\partial (\overline{\rho (\tilde{v}_i + v_i'')})}{\partial t} = \frac{\partial}{\partial t} (\overline{\bar{\rho} \tilde{v}_i} + \overline{\rho v_i''})$$

dalla quale, sfruttando la proprietà di media tale per cui $\overline{\rho Q''} = 0$, si giunge a:

$$\frac{\partial (\overline{\bar{\rho} \tilde{v}_i})}{\partial t}.$$

Si passi ora ad analizzare il secondo termine del membro di sinistra:

$$\frac{\partial (\overline{\rho (\tilde{v}_j + v_j'') (\tilde{v}_i + v_i'')})}{\partial x_j} = \frac{\partial}{\partial x_j} (\overline{\bar{\rho} \tilde{v}_i \tilde{v}_j} + \overline{\tilde{v}_j \bar{\rho} v_i''} + \overline{\tilde{v}_i \bar{\rho} v_j''} + \overline{\rho v_i'' v_j''})$$

anche in questo caso, sfruttando la medesima proprietà di media utilizzata in precedenza, i due termini prodotto tra densità e fluttuazione di velocità, in media, sono nulli e si ottiene:

$$\frac{\partial}{\partial x_j} (\overline{\bar{\rho} \tilde{v}_i \tilde{v}_j} + \overline{\rho v_i'' v_j''}).$$

Si può già notare la presenza di un nuovo termine, $\tau_{ij}^F = -\overline{\rho v_i'' v_j''}$, che prende il nome di tensore degli sforzi di Reynolds alla Favre e che è un'ulteriore incognita del sistema completo; esso dovrà essere modellato appositamente, rangan per cui sarà trattato a parte in seguito.

Si passi ora ad analizzare i termini di destra dell'equazione; il primo termine è il gradiente di pressione e si semplifica facilmente come segue sfruttando la proprietà di media della fluttuazione alla Reynolds:

$$\frac{\partial (\overline{p + p''})}{\partial x_i} = \frac{\partial (\bar{p} + \bar{p}'')}{\partial x_i} = \frac{\partial \bar{p}}{\partial x_i}.$$

Per quanto riguarda le componenti del tensore degli sforzi viscosi, $\tilde{\tau}_{ij}$, esse sono valutabili tramite l'equazione (1.12) utilizzando le componenti di velocità mediate alla Favre, poiché i termini aggiuntivi che nascono dalla scomposizione in parte media e fluttuante si possono ritenere trascurabili.

Si può ora scrivere l'equazione mediata:

$$\frac{\partial (\bar{\rho} \tilde{v}_i)}{\partial t} + \frac{\partial (\bar{\rho} \tilde{v}_i \tilde{v}_j)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} (\tilde{\mu} \tilde{\tau}_{ij} - \overline{\rho v_i'' v_j''}). \quad (1.32)$$

Equazione dell'energia mediata

Si consideri ora l'ultima equazione del sistema (1.26), ovvero l'equazione del bilancio dell'energia. Come visto nei precedenti casi la procedura per ricondursi all'equazione mediata prevede la scomposizione delle variabili e la successiva mediazione dell'equazione. I passaggi per giungere all'equazione finale sono analoghi a quelli delle equazioni precedenti, per non dilungarsi troppo nella trattazione si presenterà solo il risultato finale dell'intero processo di media:

$$\begin{aligned} \frac{\partial}{\partial t} (\bar{\rho} \tilde{E}) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{v}_j \tilde{H}) &= \frac{\partial}{\partial x_j} \left(k \frac{\partial \tilde{T}}{\partial x_j} - \overline{\rho v_j'' h''} + \overline{\mu \tau_{ij} v_i''} + \overline{\rho v_j'' K} \right) \\ &+ \frac{\partial}{\partial x_j} [\tilde{v}_i (\tilde{\mu} \tilde{\tau}_{ij} - \tau_{ij}^F)] \end{aligned} \quad (1.33)$$

Utilizzando la definizione di energia turbolenta mediata alla Favre,

$$\bar{\rho} \tilde{K} = \frac{1}{2} \overline{\rho v_i'' v_i''},$$

è possibile esprimere l'energia totale come:

$$\bar{\rho} \tilde{E} = \bar{\rho} \tilde{e} + \frac{1}{2} \tilde{v}_i \tilde{v}_i + \frac{1}{2} \overline{\rho v_i'' v_i''} = \bar{\rho} \tilde{e} + \frac{1}{2} \tilde{v}_i \tilde{v}_i + \bar{\rho} \tilde{K}$$

mentre l'entalpia totale si esprime come:

$$\bar{\rho}\tilde{H} = \bar{\rho}\tilde{h} + \frac{1}{2}\tilde{v}_i\tilde{v}_i + \frac{1}{2}\overline{\rho v_i'' v_i''} = \bar{\rho}\tilde{h} + \frac{1}{2}\tilde{v}_i\tilde{v}_i + \bar{\rho}\tilde{K}.$$

È interessante fornire un'interpretazione fisica dei singoli termini dell'equazione (1.33) come fornita da [21]:

$$\frac{\partial}{\partial x_j} \left(k \frac{\partial \tilde{T}}{\partial x_j} \right) - \text{diffusione molecolare di calore}$$

$$\frac{\partial}{\partial x_j} \left(\overline{\rho v_j'' h''} \right) - \text{trasporto turbolento di calore}$$

$$\frac{\partial}{\partial x_j} \left(\overline{\mu \tau_{ij} v_i''} \right) - \text{diffusione molecolare di } \tilde{K}$$

$$\frac{\partial}{\partial x_j} \left(\overline{\rho v_j'' K} \right) - \text{trasporto turbolento di } \tilde{K}$$

$$\frac{\partial}{\partial x_j} \left(\tilde{v}_i \tilde{\tau}_{ij} \right) - \text{lavoro fatto dal tensore degli sforzi viscosi (laminare)}$$

$$\frac{\partial}{\partial x_j} \left(\tilde{v}_i \tau_{ij}^F \right) - \text{lavoro fatto dal tensore degli sforzi di Reynolds-Favre.}$$

La diffusione molecolare e il trasporto turbolento di \tilde{K} sono spesso trascurabili, in particolare l'approssimazione è valida per flussi transonici e supersonici.

Il sistema completo delle equazioni mediate alla Reynolds-Favre, trascurando i termini di trasporto di \tilde{K} risulta dalla scrittura di (1.31), (1.32) e (1.33):

$$\left\{ \begin{array}{l} \frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{v}_j) = 0 \\ \frac{\partial}{\partial t} (\bar{\rho} \tilde{v}_i) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{v}_i \tilde{v}_j) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} (\tilde{\mu} \tilde{\tau}_{ij} - \overline{\rho v_i'' v_j''}) \\ \frac{\partial}{\partial t} (\bar{\rho} \tilde{E}) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{v}_j \tilde{H}) = \frac{\partial}{\partial x_j} \left(k \frac{\partial \tilde{T}}{\partial x_j} - \overline{\rho v_j'' h''} \right) \\ \qquad \qquad \qquad + \frac{\partial}{\partial x_j} \left[\tilde{v}_i (\tilde{\mu} \tilde{\tau}_{ij} - \tau_{ij}^F) \right] \end{array} \right. \quad (1.34)$$

Per chiudere il sistema di equazioni differenziali (1.34) è necessario fornire le sei componenti del tensore degli sforzi di Reynolds alla Favre e le tre componenti del trasporto turbolento di calore.

1.3.3 Modelli di turbolenza

Le incognite del sistema possono essere determinate tramite i modelli di turbolenza, che si suddividono in due principali tipologie: modelli basati sulla viscosità turbolenta (*Eddy-viscosity models*) e modelli che si basano sulla scrittura di equazioni differenziali che forniscono direttamente le componenti del tensore (*Reynolds-Stress Models*).

Di seguito verranno descritti i modelli *eddy-viscosity* per poi passare ad una breve introduzione sui modelli *RSM*.

Modelli *eddy-viscosity*

Il più grande contributo alla modellazione turbolenta fu fornito da Boussinesq nel 1877; la sua idea si basa sull'ipotesi di viscosità turbolenta derivante dall'osservazione del fenomeno fisico della turbolenza. Tale ipotesi può essere suddivisa in due parti; la prima è detta ipotesi *intrinseca*, afferma che la parte anisotropa del tensore degli sforzi di Reynolds

$$a_{ij} \equiv \overline{\rho u_i'' u_j''} - \frac{2}{3} \tilde{\rho} \tilde{k} \delta_{ij}$$

è determinata dal gradiente di velocità media $\frac{\partial \tilde{u}_i}{\partial x_j}$.

La seconda è detta ipotesi *specificata* e stabilisce la relazione tra a_{ij} e $\frac{\partial \tilde{u}_i}{\partial x_j}$:

$$a_{ij} = \tau_{ij}^F - \frac{2}{3} \tilde{\rho} \tilde{k} \delta_{ij} = -\mu_T \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \mu_T \frac{2}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \quad (1.35)$$

dove μ_T è la viscosità turbolenta; quest'ultima è una quantità scalare positiva che non rappresenta caratteristiche fisiche del fluido, ma è funzione delle condizioni locali dello stesso.

In generale l'ipotesi intrinseca della viscosità turbolenta non ha validità generale, ma, per flussi semplici, in cui i gradienti di velocità media e le caratteristiche turbolente evolvono lentamente, l'ipotesi risulta essere ragionevole. Nella pratica l'ipotesi viene spesso utilizzata anche al di fuori dei suoi limiti di validità, in questi casi l'approccio RANS è ancora la scelta più valida quando si è interessati a quantità macroscopiche del flusso dato il basso costo computazionale di questa tecnica.

Bisogna ora ricavare un'approssimazione del termine di trasporto turbolento di calore; una soluzione comunemente utilizzata è la seguente:

$$\frac{\partial}{\partial x_j} \left(\overline{\rho v_j'' h''} \right) = -k_T \frac{\partial \tilde{T}}{\partial x_j},$$

dove k_T è il coefficiente di conducibilità termica turbolento, definito come:

$$k_T = c_p \frac{\mu_T}{\text{Pr}_T},$$

in analogia con il caso laminare. In questa relazione Pr_T rappresenta il numero di Prandtl turbolento, che in generale si ritiene costante nel fluido e per l'aria vale $\text{Pr}_T = 0.9$.

Applicando l'approccio appena descritto al sistema di equazioni differenziali (1.34), il tensore degli sforzi turbolenti è sostituito con una forma del tutto analoga al caso laminare, ovvero:

$$-\overline{\rho v_i'' v_j''} = \mu_T \tilde{\tau}_{ij}.$$

A questo punto l'ultimo termine del membro di destra della seconda equazione della (1.34) può essere riscritto come:

$$\frac{\partial}{\partial x_j} (\tilde{\mu} \tilde{\tau}_{ij} - \overline{\rho v_i'' v_j''}) = \frac{\partial}{\partial x_j} (\mu_{\text{tot}} \tilde{\tau}_{ij}) \quad (1.36)$$

dove il coefficiente di viscosità dinamica μ_{tot} è dato dalla somma tra il coefficiente laminare e quello turbolento:

$$\mu_{\text{tot}} = \mu + \mu_T.$$

Allo stesso modo il coefficiente di conducibilità termica k si riscrive come:

$$k_{\text{tot}} = k + k_T = c_p \left(\frac{\mu}{\text{Pr}_L} + \frac{\mu_T}{\text{Pr}_T} \right).$$

Data la completa somiglianza alle equazioni di Navier-Stokes non mediate (1.26) è possibile concludere che la loro versione mediata si può ottenere considerando le incognite del problema come valori medi e sostituendo il coefficiente di viscosità e quello di conducibilità termica con la somma della loro parte laminare più quella turbolenta, non più uniforme in tutto il campo di moto, ottenendo il seguente sistema di equazioni mediate:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_c - \nabla \cdot (\mu_{\text{tot}}^1 \mathbf{F}^{v1} + \mu_{\text{tot}}^2 \mathbf{F}^{v2}) = \mathbf{Q}. \quad (1.37)$$

dove in questo caso le variabili μ^1 e μ^2 sono:

$$\mu_{\text{tot}}^1 = \mu + \mu_T \text{ e } \mu_{\text{tot}}^2 = \frac{\mu}{\text{Pr}_L} + \frac{\mu_T}{\text{Pr}_T}$$

Questa tecnica, almeno dal punto di vista ingegneristico, è molto comoda poiché richiede solo la determinazione della viscosità turbolenta. Quindi

l'ipotesi di Boussinesq è la base di tutte le tecniche del prim'ordine di chiusura delle equazioni turbolente. Come già detto le limitazioni sono molteplici, ma i risultati, al di fuori del campo di validità delle ipotesi, possono essere migliorati tramite correzioni o utilizzando modelli di viscosità turbolenta non lineare, più complessi e che in questo lavoro non verranno trattati.

I modelli *eddy-viscosity* si suddividono in tre categorie:

- modelli algebrici,
- modelli ad una equazione differenziale,
- modelli a due equazioni differenziali.

Di seguito viene descritto un particolare modello ad una equazione, il modello *Spalart-Allmaras*, mentre per le altre classi di modelli si può fare riferimento a [16, 3].

Modello ad una equazione di Spalart-Allmaras

Questo modello utilizza un'equazione di trasporto per la viscosità turbolenta $\tilde{\nu}$; l'equazione è stata sviluppata basandosi su considerazioni empiriche, analisi dimensionale e invarianza galileiana. Nell'equazione sono presenti tre funzioni empiriche e ben otto coefficienti empirici, che sono stati tarati basandosi su risultati per il *mixing-layer* 2D, scie e strato limite su lamina piana. Il modello risulta accurato se utilizzato con flussi turbolenti con gradiente avverso ed è ottimizzato per flussi turbolenti su ali e scie ad esse connesse. Ha particolari peculiarità anche dal punto di vista numerico, infatti è un'equazione locale che può essere facilmente implementata su strutture multi-blocco o su griglie non strutturate. Inoltre risulta essere un modello robusto, converge velocemente alla soluzione stazionaria e non richiede elevata risoluzione nelle zone vicino parete.

La forma differenziale dell'equazione, scritta in notazione tensoriale, è la seguente:

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + \frac{\partial}{\partial x_j} (\tilde{\nu} \tilde{v}_j) = & C_{b1} (1 - f_{t2}) \tilde{S} \tilde{\nu} \\ & + \frac{1}{\sigma} \left\{ \frac{\partial}{\partial x_j} \left[(\nu_L + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right] + C_{b2} \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} \right\}. \quad (1.38) \\ & - \left[C_{w1} f_{w1} - \frac{C_{b1}}{\kappa^2} f_{t2} \right] \left(\frac{\tilde{\nu}}{d} \right)^2 \end{aligned}$$

I termini di destra dell'equazione rappresentano rispettivamente: produzione di viscosità turbolenta, diffusione e dissipazione, distruzione di turbolenza

vicino parete e smorzamento della produzione. Inoltre $\nu_L = \mu_L/\bar{\rho}$ è la viscosità cinematica laminare e d rappresenta la distanza dalla parete più vicina. La viscosità dinamica turbolenta si ricava dalla relazione:

$$\mu_T = f_{v1}\bar{\rho}\tilde{\nu},$$

mentre il termine di produzione si ricava dalla seguente formula:

$$\begin{aligned} \tilde{S} &= f_{v3}S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \\ f_{v1} &= \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad f_{v2} = \left(1 + \frac{\chi}{C_{v2}}\right)^{-3} \\ f_{v3} &= \frac{(1 + \chi f_{v1})}{\max(\chi, 0.001)} (1 - f_{v2}), \quad \chi = \frac{\tilde{\nu}}{\nu_L} \end{aligned} \quad (1.39)$$

dove S è definita da:

$$S = \sqrt{2\Omega_{ij}\Omega_{ij}} = \sqrt{\frac{1}{2} \left(\frac{\partial \tilde{v}_i}{\partial x_j} - \frac{\partial \tilde{v}_j}{\partial x_i} \right)^2}.$$

Le altre funzioni empiriche dell'equazione sono:

$$\begin{aligned} f_w &= g \left(\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right)^{\frac{1}{6}}, \\ g &= r + C_{w2} (r^6 - r), \quad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}. \end{aligned}$$

Infine le costanti sono:

$$\begin{aligned} C_{b1} &= 0.1355, \quad C_{b2} = 0.622 \\ C_{v1} &= 7.1, \quad C_{v2} = 5, \quad \sigma = \frac{2}{3}, \quad \kappa = 0.41 \\ C_{w1} &= \frac{C_{b1}}{\kappa^2} + \frac{(1 + C_{b2})}{\sigma}, \quad C_{w2} = 0.3, \quad c_{w3} = 2. \end{aligned}$$

Per poter scrivere le equazioni in un codice a volumi finiti è necessario ricorrersi alla forma integrale, che è:

$$\frac{\partial}{\partial t} \int_{\Omega} \tilde{\nu} d\Omega + \oint_{\partial\Omega} (F_{c,T} - F_{v,T}) dS = \int_{\Omega} Q_T d\Omega, \quad (1.40)$$

dove $F_{c,T}$ è il flusso convettivo ed è definito da:

$$F_{c,T} = \tilde{\nu} \tilde{\mathbf{v}} \cdot \mathbf{n},$$

solitamente discretizzato tramite schema upwind del primo ordine, mentre il flusso viscoso è:

$$F_{v,T} = \boldsymbol{\tau}_T \cdot \mathbf{n},$$

dove $\boldsymbol{\tau}_T$ è il vettore degli sforzi viscosi normali:

$$\tau_{T_{ii}} = \frac{1}{\sigma} (\nu_L + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_i}.$$

Infine il termine sorgente è:

$$Q_T = C_{b1} (1 - f_{t2}) \tilde{S} \tilde{\nu} + C_{b2} \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} - \left[C_{w1} f_{w1} - \frac{C_{b1}}{\kappa^2} f_{t2} \right] \left(\frac{\tilde{\nu}}{d} \right)^2.$$

Sfruttando il teorema di Gauss è possibile riscrivere il secondo integrale dell'equazione (1.40) in integrale di volume e ottenere l'equazione differenziale:

$$\frac{\partial \tilde{\nu}}{\partial t} + \nabla \cdot \mathbf{T}^{cv} - Q_T = 0 \quad (1.41)$$

dove \mathbf{T}^{cv} rappresenta il flusso viscoso e convettivo, ovvero:

$$\mathbf{T}^{cv} = -\frac{\nu_L + \tilde{\nu}}{\sigma} \nabla \tilde{\nu} + \tilde{\mathbf{v}} \tilde{\nu}.$$

Nell'equazione è presente la variabile d che rappresenta la distanza dalla parete più vicina; tale variabile è comune nei modelli di turbolenza ed è soluzione dell'equazione di Eikonal:

$$|\nabla d|^2 - 1 = 0 \quad (1.42)$$

in questo caso la soluzione d è proprio la distanza dalla superficie $\partial\Omega$.

Solitamente si impone come condizione iniziale $\tilde{\nu} = 3 \div 5\nu_L$, la stessa condizione si impone solitamente sul contorno di *inflow*; sul contorno di *outflow* la variabile turbolenta viene semplicemente estrolata dal dominio. Sulle pareti solide invece si impone $\tilde{\nu}$ nulla e di conseguenza: $\mu_T = 0$.

Modelli *RSM*

Nei modelli di turbolenza *RSM*, introdotti per la prima volta da Launder nel 1975, vengono risolte delle equazioni di trasporto modellate per ogni componente del tensore degli sforzi τ_{ij}^F e solitamente si aggiunge anche un'equazione aggiuntiva per la dissipazione ε . Di conseguenza, l'ipotesi di viscosità turbolenta non è più necessaria, eliminando uno dei difetti più grandi dei modelli visti in precedenza. Per contro, un aspetto negativo è sicuramente l'elevato

costo computazionale legato al fatto di dover risolvere diverse equazioni differenziali aggiuntive.

L'equazione del trasporto del tensore degli sforzi di Reynolds si ottiene dalle equazioni di Navier-Stokes ed è:

$$\bar{u}_k \frac{\partial \tau_{ij}^F}{\partial x_k} = - \frac{\partial}{\partial x_k} T_{ijk} + \mathcal{P}_{ij} + \mathcal{R}_{ij} - \varepsilon_{ij}. \quad (1.43)$$

In questa equazione il termine di convezione del flusso medio e il tensore di produzione, \mathcal{P}_{ij} , sono in forma chiusa; devono invece essere forniti modelli per il tensore di dissipazione, ε_{ij} , per il tensore legato ai processi distributivi, \mathcal{R}_{ij} e per il flusso degli sforzi di Reynolds, T_{ijk} .

La trattazione relativa alla chiusura di questa equazione non viene qui riportata, ma è possibile approfondire l'argomento seguendo il Capitolo 11 del [16].

Capitolo 2

Ottimizzazione geometrica

L'ottimizzazione geometrica in aerodinamica è stato un importante campo di ricerca per diversi anni, ma i primi lavori in quest'area di ricerca avevano applicazioni limitate a causa della scarsa potenza di calcolo a disposizione [2]. Grazie all'avvento di algoritmi e hardware avanzati è stato possibile applicare la teoria a casi più complessi. Tra i vari metodi attualmente utilizzati i metodi *gradient-based* permettono di minimizzare una funzione obiettivo J scelta a priori; i gradienti del funzionale J rispetto alle variabili di progetto vengono sfruttati per modificare queste ultime, quindi modificare la geometria, in modo da ridurre il valore della funzione obiettivo per raggiungere un minimo locale. È dunque una questione importante il modo in cui vengono calcolati i gradienti della funzione obiettivo, che prendono il nome di derivate di sensitività o sensitività aerodinamica.

A questo proposito sono diverse le tecniche per il calcolo della sensitività, ma di particolare interesse sono i metodi aggiunti [10]. Il primo utilizzo di tali tecniche nell'ambito fluidodinamico è da attribuirsi a Pironneau [15], mentre il primo utilizzo in CFD, per scopi aeronautici, è attribuito a Jameson, che sfruttò la teoria del controllo ottimo per sviluppare quelli che prendono il nome di *optimal shape methods*.

Nei metodi aggiunti la funzione obiettivo è aumentata con le equazioni di governo (i vincoli del problema) tramite l'utilizzo dei moltiplicatori di Lagrange; da questo procedimento è chiaro il punto di vista del controllo ottimo. Tale tecnica è particolarmente utile quando si lavora con un gran numero di variabili di progetto perché il calcolo aggiunto ha un costo computazionale comparabile a quello del calcolo della soluzione delle equazioni di governo (simulazione diretta), mentre il calcolo della sensitività ha un costo trascurabile. Questo è il punto di forza del metodo aggiunto, che lo rende preferibile ad altri approcci, come ad esempio le differenze finite. Quest'ultimo metodo richiede infatti di deformare una variabile di design alla volta ed eseguire una simulazione diretta per ognuna di esse; la sensitività calcolata con questo

metodo è esatta, ma in presenza di un gran numero di variabili il metodo è inapplicabile o quanto meno molto oneroso.

Il ciclo completo di ottimizzazione è schematizzato in Figura 2.1; si può osservare come per il calcolo della sensitività siano necessarie la soluzione diretta e aggiunta del flusso. Questa è una questione importante del metodo infatti, come sarà esposto nelle sezioni successive, la sensitività è funzione sia delle variabili dirette sia di quelle aggiunte del flusso.

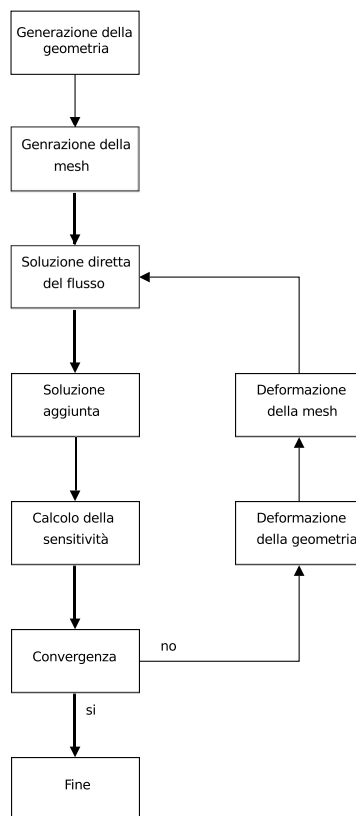


Figura 2.1: Ciclo di ottimizzazione geometrica.

Di seguito si esporrà la teoria del metodo aggiunto per flussi turbolenti comprimibili, così come è implementata in SU^2 e due possibili scelte delle variabili di design che permettono la deformazione della geometria. Inoltre la convergenza e la deformazione della geometria sono gestite da un algoritmo di ottimizzazione, che nel caso di SU^2 è l'SLSQP, il cui funzionamento verrà spiegato anch'esso nelle sezioni successive.

2.1 Metodo aggiunto

La principale distinzione tra i metodi aggiunti è tra discreti e continui [10]; la differenza tra i due approcci sta nell'ordine sequenziale in cui si esegue la discretizzazione della funzione obiettivo e il calcolo della sua variazione. Nell'approccio discreto la funzione obiettivo aumentata viene prima discretizzata e successivamente si calcola la sua variazione, mentre in quello continuo prima si calcolano le variazioni del funzionale e successivamente si discretizzano le equazioni. L'approccio continuo permette un'interpretazione più chiara delle equazioni aggiunte e delle condizioni al contorno; inoltre permette di scegliere gli schemi di discretizzazione delle equazioni, che possono anche essere diversi da quelli delle equazioni primarie. L'approccio discreto, invece, fornisce il gradiente esatto della funzione obiettivo discretizzata e assicura che il processo di ottimizzazione converga, tale tecnica però può richiedere elevata memoria. In generale la sensitività calcolata con i due metodi può differire.

Nel codice SU^2 sono implementati entrambi i metodi, ma di seguito sarà proposta la procedura per ricavare le equazioni aggiunte dell'approccio continuo, che di seguito sarà semplicemente chiamato aggiunto, e le relative condizioni al contorno [4]. L'approccio discreto è implementato nel codice, ma non può essere utilizzato all'interno del processo di ottimizzazione.

A tale proposito si consideri il sistema completo delle equazioni di Navier-Stokes turbolente (1.25) in un dominio $\Omega \subset \mathbb{R}^3$ delimitato da contorni disconnessi divisi in *farfield* Γ_∞ e una parete adiabatica S . Inoltre si consideri il caso di flusso stazionario con assenza di sorgenti esterne, il sistema che si ottiene è il seguente:

$$\begin{cases} \mathbf{R}_U(\mathbf{U}, \tilde{\nu}) = \nabla \cdot \underline{\underline{F}}^c - \nabla \cdot (\mu_{tot}^1 \underline{\underline{F}}^{v1} + \mu_{tot}^2 \underline{\underline{F}}^{v2}) = 0 & \text{in } \Omega \\ \mathbf{v} = 0 & \text{su } S \\ \partial_n T = 0 & \text{su } S \\ (W)_+ = W_\infty & \text{su } \Gamma_\infty \end{cases} \quad (2.1)$$

dove l'ultima equazione rappresenta una condizione che simula il comportamento del fluido all'infinito.

Come già discusso nella Sezione 1.3, si introduce l'equazione del modello di turbolenza Spalart-Allmaras (1.41), con relative condizioni al contorno, per completare la chiusura delle equazioni turbolente:

$$\begin{cases} R_{\tilde{\nu}}(\mathbf{U}, \tilde{\nu}, d) = \frac{\partial \tilde{\nu}}{\partial t} + \nabla \cdot \mathbf{T}^{cv} - Q_T = 0 & \text{in } \Omega \\ \tilde{\nu} = 0 & \text{su } S \\ \tilde{\nu}_\infty = \sigma_\infty \nu_\infty & \text{su } \Gamma_\infty \end{cases} \quad (2.2)$$

si noti che nell'equazione appena scritta c'è una dipendenza esplicita dalla distanza dalla parete d , che è soluzione dell'equazione di Eikonal (1.42):

$$\begin{cases} R_d(d) = |\nabla d|^2 - 1 = 0 & \text{in } \Omega \\ d = 0 & \text{su } S. \end{cases} \quad (2.3)$$

I sistemi scritti in precedenza descrivono completamente il comportamento termo-fluidodinamico del flusso.

Per i problemi di ottimizzazione geometrica è fondamentale la definizione di una funzione obiettivo; in generale si può definire la funzione su qualsiasi superficie fisica e non. Inoltre è possibile dimostrare che le equazioni di Navier-Stokes permettono, a priori, di scegliere come funzione obiettivo una qualsiasi componente della forza totale esercitata dal flusso sulla superficie (dove per forza totale si intende sia la componente di pressione che quella viscosa) o la distribuzione di temperatura o ancora il flusso di calore. Nell'articolo di Castro et al. [5] è mostrato come per flussi stazionari comprimibili viscosi, ma in generale il ragionamento può essere esteso anche a flussi incomprimibili instazionari, sia possibile omettere il termine viscoso, almeno formalmente, per qualsiasi funzione di costo dipendente solamente dalla pressione.

A tale proposito si consideri quindi una funzione obiettivo generica J :

$$J(S) = \oint_S j(\mathbf{f}, T, \partial_n \tilde{v}, \mathbf{n}) ds$$

dove \mathbf{f} è la forza totale data da:

$$\mathbf{f} = (f_1, f_2, f_3) = p\mathbf{n} - \underline{\underline{\sigma}} \cdot \mathbf{n}, \quad \underline{\underline{\sigma}} = \mu_{\text{tot}}^1 \underline{\underline{\tilde{T}}}.$$

Si può dimostrare come, per fluidi viscosi instazionari su pareti solide con condizione di non slittamento, valga la seguente relazione [5]:

$$(\mathbf{n} \cdot \underline{\underline{\sigma}} \cdot \mathbf{n}) = \frac{4}{3}\mu \nabla \cdot \mathbf{v} = -\frac{4}{3}\mu \frac{1}{\rho} \frac{\partial \rho}{\partial t},$$

si può quindi concludere che il termine $\mathbf{n} \cdot \underline{\underline{\sigma}} \cdot \mathbf{n}$ su S è nullo per flussi stazionari. Sfruttando il fatto che $\mathbf{n} \cdot \underline{\underline{\sigma}} \cdot \mathbf{n} = 0$ è possibile scrivere:

$$\mathbf{f} \cdot \mathbf{n} = p\mathbf{n} \cdot \mathbf{n} - \mathbf{n} \cdot \underline{\underline{\sigma}} \cdot \mathbf{n} = p \text{ su } S \quad (2.4)$$

di conseguenza funzioni che dipendono solo dalla pressione sono formalmente ammesse.

Come già detto, l'approccio aggiunto prevede di scrivere una funzione costo aumentata con le equazioni di governo tramite i moltiplicatori di Lagrange:

$$\begin{aligned} \mathcal{J}(S) = & \oint_S j(\mathbf{f}, T, \partial_n \tilde{\nu}, \mathbf{n}) ds \\ & + \int_{\Omega} (\Psi_U \cdot \mathbf{R}_U(\mathbf{U}, \tilde{\nu}) + \Psi_{\tilde{\nu}} R_{\tilde{\nu}}(\mathbf{U}, \tilde{\nu}, d) + \Psi_d R_d(d)) d\Omega \end{aligned} \quad (2.5)$$

dove $\Psi_U = (\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5)$, $\Psi_{\tilde{\nu}}$ e Ψ_d sono i moltiplicatori di Lagrange, anche detti *variabili aggiunte*.

Si consideri ora una perturbazione arbitraria, ma piccola, della superficie S che può essere descritta come segue:

$$\delta \mathbf{x} = \alpha \mathbf{n} + \beta \mathbf{t},$$

dove \mathbf{n} è il vettore normale ad S che punta verso l'esterno del dominio fluido e \mathbf{t} è il vettore tangente a S .

Nell'articolo di Castro et al. [5] si mostra come in realtà una deformazione sufficientemente piccola di una superficie possa essere descritta da una deformazione normale alla superficie stessa, mentre la deformazione tangenziale si riduce ad una semplice riparametrizzazione della stessa. Per questo motivo la relazione precedente si può ridurre trascurando il termine tangenziale. Per piccole deformazioni possiamo scrivere:

$$\begin{cases} \delta \mathbf{n} = -\nabla_S \alpha \\ \delta(ds) = -2H_m \alpha ds \end{cases} \quad (2.6)$$

dove H_m è la curvatura media di S calcolata come $(\kappa_1 + \kappa_2)/2$ e (κ_1, κ_2) sono le curvature in due direzioni ortogonali alla superficie; mentre ∇_S rappresenta l'operatore di derivata tangenziale alla superficie, quindi in questo caso $\nabla_S \alpha$ è un vettore tangente in ogni punto a S .

Si ipotizzi ora di avere una soluzione del flusso $\mathbf{U}, \tilde{\nu}, d$; allora la variazione del funzionale J , a seguito di una deformazione di S , può essere valutata come:

$$\begin{aligned} \delta \mathcal{J} = & \oint_S \delta j(\mathbf{f}, T, \partial_n \tilde{\nu}, \mathbf{n}) ds + \oint_{\delta S} j(\mathbf{f}, T, \partial_n \tilde{\nu}, \mathbf{n}) ds \\ & + \int_{\Omega} (\Psi_U \delta \mathbf{R}_U(\mathbf{U}, \tilde{\nu}) + \Psi_{\tilde{\nu}} \delta R_{\tilde{\nu}}(\mathbf{U}, \tilde{\nu}, d) + \Psi_d \delta R_d(d)) d\Omega \end{aligned} \quad (2.7)$$

dove $\delta \mathbf{R}_U, \delta R_{\tilde{\nu}}$ e δR_d rappresentano le variazioni delle equazioni di governo del fluido e δj la variazione del funzionale j rispetto alle variabili del flusso. Si esplicitino ora i termini dell'equazione precedente; il primo termine può

essere riscritto come segue:

$$\begin{aligned} \delta j(\mathbf{f}, T, \partial_n \tilde{\nu}, \mathbf{n}) &= \frac{\partial j}{\partial f_i} \delta f_i + \frac{\partial j}{\partial T} \delta T + \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \delta (\partial_n \tilde{\nu}) = \\ &= \frac{\partial j}{\partial \mathbf{f}} \cdot (\delta p \mathbf{n} + p \delta \mathbf{n} - \delta \underline{\underline{\sigma}} \mathbf{n} - \underline{\underline{\sigma}} \delta \mathbf{n}) + \frac{\partial j}{\partial T} \delta T + \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \delta (\partial_n \tilde{\nu}) = \\ &= \frac{\partial j}{\partial \mathbf{f}} \cdot (\delta p \mathbf{n} - \delta \underline{\underline{\sigma}} \mathbf{n}) + \frac{\partial j}{\partial T} \delta T + \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \delta (\partial_n \tilde{\nu}) - \left(\frac{\partial j}{\partial \mathbf{f}} p - \frac{\partial j}{\partial \mathbf{f}} \cdot \underline{\underline{\sigma}} \right) \cdot \delta \mathbf{n}. \end{aligned}$$

Sfruttando la prima equazione delle (2.6) si può riscrivere la variazione del vettore normale, ottenendo così la scrittura del primo termine del membro di destra della (2.7):

$$\begin{aligned} \delta j(\mathbf{f}, T, \partial_n \tilde{\nu}, \mathbf{n}) &= \frac{\partial j}{\partial \mathbf{f}} \cdot (\delta p \mathbf{n} - \delta \underline{\underline{\sigma}} \mathbf{n}) + \frac{\partial j}{\partial T} \delta T \\ &+ \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \delta (\partial_n \tilde{\nu}) - \left(\frac{\partial j}{\partial \mathbf{f}} p - \frac{\partial j}{\partial \mathbf{f}} \cdot \underline{\underline{\sigma}} \right) \cdot \nabla_S \alpha. \end{aligned} \quad (2.8)$$

Facendo riferimento ai lavori [11, 5], si può riscrivere il secondo termine della (2.7), che prende il nome di termine *geometrico*, come segue:

$$\oint_{\delta S} j(\mathbf{W}, \mathbf{n}) ds = \oint_S \frac{\partial j}{\partial \mathbf{W}} (\delta \mathbf{x} \cdot \nabla \mathbf{W}) ds + \oint_S \frac{\partial j}{\partial \mathbf{n}} \cdot \delta \mathbf{n} ds + \oint_S j \delta ds$$

dove \mathbf{W} è il vettore delle variabili del flusso da cui dipende la funzione j . Nel caso in esame la relazione precedente può essere scritta come segue:

$$\begin{aligned} \oint_{\delta S} j(\mathbf{f}, T, \partial_n \tilde{\nu}, \mathbf{n}) ds &= \oint_S \frac{\partial j}{\partial f_i} (\delta \mathbf{x} \cdot \nabla f_i) ds + \oint_S \frac{\partial j}{\partial T} (\delta \mathbf{x} \cdot \nabla T) ds \\ &+ \oint_S \frac{\partial j}{\partial (\partial_n \tilde{\nu})} (\delta \mathbf{x} \cdot \nabla (\partial_n \tilde{\nu})) ds \\ &+ \oint_S \frac{\partial j}{\partial \mathbf{n}} \cdot \delta \mathbf{n} ds + \oint_S j \delta ds \end{aligned}$$

da cui sostituendo $\delta \mathbf{x}$ con la deformazione normale e $\delta(ds)$ seguendo la seconda relazione delle (2.6), si ottiene:

$$\oint_{\delta S} j(\mathbf{f}, T, \partial_n \tilde{\nu}, \mathbf{n}) ds = \oint_S \left(\frac{\partial j}{\partial f_i} \partial_n f_i + \frac{\partial j}{\partial T} \partial_n T + \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \partial_n^2 \tilde{\nu} - \frac{\partial j}{\partial \mathbf{n}} \cdot \nabla_S - 2H_m j \right) \alpha ds, \quad (2.9)$$

con $\partial_n = \mathbf{n} \cdot \nabla$.

Nell'equazione (2.8) sono presenti i termini di variazione delle variabili del

flusso $\delta p \mathbf{n} - \delta \underline{\underline{\sigma}} \cdot \mathbf{n}$, δT , e $\delta(\partial_n \tilde{\nu})$ che possono essere calcolati a partire dalle equazioni linearizzate del flusso:

$$\begin{cases} \delta \mathbf{R}_U(\mathbf{U}, \tilde{\nu}) = \frac{\partial \mathbf{R}_U}{\partial \mathbf{U}} \delta \mathbf{U} + \frac{\partial \mathbf{R}_U}{\partial \tilde{\nu}} \delta \tilde{\nu} = 0 & \text{in } \Omega \\ \delta \mathbf{v} = -\partial_n \mathbf{v} \delta S & \text{su } S \\ \partial_n(\delta T) = (\nabla T) \cdot \nabla_S(\delta S) - \partial_n^2 T \delta S & \text{su } S \\ (\delta W)_+ = 0 & \text{su } \Gamma_\infty \end{cases} \quad (2.10)$$

$$\begin{cases} \delta R_{\tilde{\nu}}(\mathbf{U}, \tilde{\nu}, d) = \frac{\partial R_{\tilde{\nu}}}{\partial \mathbf{U}} \delta \mathbf{U} + \frac{\partial R_{\tilde{\nu}}}{\partial \tilde{\nu}} \delta \tilde{\nu} + \frac{\partial R_{\tilde{\nu}}}{\partial d} \delta d = 0 & \text{in } \Omega \\ \delta \tilde{\nu} = -\partial_n \tilde{\nu} \delta S & \text{su } S \\ \delta \tilde{\nu} = \sigma_\infty \delta \nu & \text{su } \Gamma_\infty \end{cases} \quad (2.11)$$

$$\begin{cases} \delta R_d(d) = \nabla d \cdot \nabla \delta d = 0 & \text{in } \Omega \\ \delta d = \alpha & \text{su } S \end{cases} \quad (2.12)$$

Nella condizione al contorno dell'equazione linearizzata di Eikonal (2.12) è stato sfruttato il fatto che $\partial_n d = -1$, di conseguenza $\delta d = -\partial_n \delta \alpha = \alpha$. Per la linearizzazione delle equazioni di Navier-Stokes (2.10) e dell'equazione di Spalart-Allmaras (2.11) si faccia riferimento al lavoro di Bueno-Orovio et al. [4].

A questo punto gli integrali sul dominio Ω nell'equazione (2.7) possono essere eliminati sfruttando l'integrazione per parti e introducendo gli operatori aggiunti. Tale procedura è ben sviluppata nel riferimento [4], di conseguenza verranno solo proposti i risultati. L'integrazione per parti produce anche dei termini integrali su S che si aggiungono a quelli già presenti nella relazione, fornendo così le condizioni al contorno delle equazioni aggiunte. Di conseguenza gli ultimi tre termini della variazione del funzionale aggiunto nell'equazione (2.7), sfruttando le equazioni linearizzate sono:

$$\begin{aligned} & \int_{\Omega} \Psi_U \cdot \overbrace{\left(\frac{\partial \mathbf{R}_U}{\partial \mathbf{U}} \delta \mathbf{U} + \frac{\partial \mathbf{R}_U}{\partial \tilde{\nu}} \delta \tilde{\nu} \right)}^{\delta \mathbf{R}_U} d\Omega + \int_{\Omega} \Psi_{\tilde{\nu}} \overbrace{\left(\frac{\partial R_{\tilde{\nu}}}{\partial \mathbf{U}} \delta \mathbf{U} + \frac{\partial R_{\tilde{\nu}}}{\partial \tilde{\nu}} \delta \tilde{\nu} + \frac{\partial R_{\tilde{\nu}}}{\partial d} \delta d \right)}^{\delta R_{\tilde{\nu}}} d\Omega \\ & + \int_{\Omega} \Psi_d \underbrace{\left(\frac{\partial R_d}{\partial d} \delta d \right)}_{\delta R_d} d\Omega = \int_{\Omega} (A_U^U \Psi_U + A_{\tilde{\nu}}^U \Psi_{\tilde{\nu}}) \delta \mathbf{U} d\Omega + \int_{\Omega} (A_U^{\tilde{\nu}} \Psi_U + A_{\tilde{\nu}}^{\tilde{\nu}} \Psi_{\tilde{\nu}}) \delta \tilde{\nu} d\Omega \\ & + \int_{\Omega} (A_{\tilde{\nu}}^d \Psi_{\tilde{\nu}} + A_d^d \Psi_d) \delta d d\Omega + \oint_S B_S ds \end{aligned} \quad (2.13)$$

dove A_U^U , $A_{\tilde{\nu}}^U$, $A_U^{\tilde{\nu}}$, $A_{\tilde{\nu}}^{\tilde{\nu}}$, $A_{\tilde{\nu}}^d$ e A_d^d sono gli operatori aggiunti mentre B_S rappresenta i termini integrali di superficie che derivano dall'integrazione per

parti:

$$\begin{aligned}
 \oint_S B_S ds &= - \oint_S \boldsymbol{\varphi} \cdot (\delta p \mathbf{n} - \delta \underline{\underline{\sigma}} \cdot \mathbf{n}) ds \\
 &\quad - \oint_S (\mathbf{g}_1 \cdot \boldsymbol{\varphi} + g_2 \partial_n \psi_5 + \psi_{\tilde{\nu}} g_5) \delta T ds - \oint_S g_3 \psi_{\tilde{\nu}} \delta (\partial_n \tilde{\nu}) ds \\
 &\quad + \oint_S \Psi_{\tilde{\nu}} g_4 \delta p ds - \oint_S \Psi_d \alpha ds + \oint_S \hat{g} \alpha ds
 \end{aligned}$$

dove $\boldsymbol{\varphi} = (\Psi_2, \Psi_3, \Psi_4)$, \mathbf{g}_1 e g_i con $i = 1, 2, 3, 4, 5$ sono funzioni che non dipendono dalle variabili aggiunte, mentre \hat{g} è una funzione che non dipende dalla variabile aggiunta Ψ_d , ancora una volta si fa riferimento al lavoro [4]. Si osservi che è possibile riscrivere la variazione di pressione δp eseguendo una linearizzazione della relazione (2.4) come segue:

$$\delta p = \delta (p \mathbf{n} \cdot \mathbf{n} - \mathbf{n} \cdot \underline{\underline{\sigma}} \cdot \mathbf{n}) = \mathbf{n} \cdot (\delta p \mathbf{n} - \delta \underline{\underline{\sigma}} \cdot \mathbf{n}) - 2 (p \mathbf{n} - \mathbf{n} \cdot \underline{\underline{\sigma}}) \cdot \nabla_S \alpha \text{ su } S$$

quindi si può riscrivere il termine di superficie:

$$\begin{aligned}
 \oint_S B_S ds &= \oint_S (-\boldsymbol{\varphi} + \Psi_{\tilde{\nu}} g_4) \cdot (\delta p \mathbf{n} - \delta \underline{\underline{\sigma}} \cdot \mathbf{n}) ds \\
 &\quad - \oint_S (\mathbf{g}_1 \cdot \boldsymbol{\varphi} + g_2 \partial_n \psi_5 + \psi_{\tilde{\nu}} g_5) \delta T ds - \oint_S g_3 \psi_{\tilde{\nu}} \delta (\partial_n \tilde{\nu}) ds \\
 &\quad - \oint_S \Psi_d \alpha ds - \oint_S \hat{g} \alpha ds - 2 \oint_S \Psi_{\tilde{\nu}} g_4 (p \mathbf{n} - \mathbf{n} \cdot \underline{\underline{\sigma}}) \cdot \nabla_S \alpha ds
 \end{aligned} \tag{2.14}$$

Combinando le equazioni (2.8) (2.9) (2.13), la variazione del funzionale sarà:

$$\begin{aligned}
 \delta \mathcal{J} &= \oint_S \left(\frac{\partial j}{\partial \mathbf{f}} - \boldsymbol{\varphi} + \Psi_{\tilde{\nu}} g_4 \right) \cdot (\delta p - \delta \underline{\underline{\sigma}} \cdot \mathbf{n}) ds \\
 &\quad + \oint_S \left(\frac{\partial j}{\partial T} - \mathbf{g}_1 \cdot \boldsymbol{\varphi} - g_2 \partial_n \Psi_5 - \Psi_{\tilde{\nu}} g_5 \right) \delta T ds \\
 &\quad + \oint_S \left(\frac{\partial j}{\partial \partial_n \tilde{\nu}} - g_3 \Psi_{\tilde{\nu}} \right) \delta (\partial_n \tilde{\nu}) ds \\
 &\quad + \oint_S \left(-\Psi_d - \hat{g} + \frac{\partial j}{\partial f_i} \partial_n f_i + \frac{\partial j}{\partial T} \partial_n T + \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \partial_n^2 \tilde{\nu} - 2H_m j \right) \alpha ds \\
 &\quad + \oint_S \left(-2\Psi_{\tilde{\nu}} g_4 (p \mathbf{n} - \underline{\underline{\sigma}} \cdot \mathbf{n}) - \frac{\partial j}{\partial \mathbf{f}} p + \frac{\partial j}{\partial \underline{\underline{\sigma}}} - \frac{\partial j}{\partial \mathbf{n}} \right) \cdot \nabla_S \alpha ds \\
 &\quad + \int_{\Omega} (A_U^U \boldsymbol{\Psi}_U + A_{\tilde{\nu}}^U \Psi_{\tilde{\nu}}) \delta U d\Omega + \int_{\Omega} (A_U^{\tilde{\nu}} \boldsymbol{\Psi}_U + A_{\tilde{\nu}}^{\tilde{\nu}} \Psi_{\tilde{\nu}}) \delta \tilde{\nu} d\Omega \\
 &\quad + \int_{\Omega} (A_{\tilde{\nu}}^d \Psi_{\tilde{\nu}} + A_d^d \Psi_d) \delta d d\Omega
 \end{aligned} \tag{2.15}$$

Per eliminare i termini integrati nel dominio Ω dalla scrittura della variazione del funzionale nell'equazione (2.15) si impone che le variabili aggiunte soddisfino le seguenti equazioni:

$$A_U^U \Psi_U + A_\nu^U \Psi_\nu = 0 \quad (2.16)$$

$$A_U^{\tilde{\nu}} \Psi_U + A_\nu^{\tilde{\nu}} \Psi_\nu = 0 \quad (2.17)$$

$$A_\nu^d \Psi_\nu + A_d^d \Psi_d = 0 \quad (2.18)$$

Allo stesso modo i termini di superficie senza dipendenza esplicita da α , ovvero i primi tre integrali di (2.15), possono essere eliminati scegliendo le seguenti condizioni al contorno per le variabili aggiunte:

$$\Psi_\nu = \frac{1}{g_3} \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \quad \text{su } S \quad (2.19)$$

$$\boldsymbol{\varphi} = \frac{\partial j}{\partial \mathbf{f}} + \Psi_\nu g_4 \mathbf{n} \quad \text{su } S \quad (2.20)$$

$$\partial_n \Psi_5 = \frac{1}{g_2} \left(\frac{\partial j}{\partial T} - \mathbf{g}_1 \cdot \boldsymbol{\varphi} - \Psi_\nu g_5 \right) \quad \text{su } S \quad (2.21)$$

$$\Psi_d = 0 \quad \text{su } S \quad (2.22)$$

Si noti che le condizioni al contorno devono essere scritte in questo ordine poichè il valore di Ψ_ν nelle equazioni (2.20) e (2.21) è ottenuto dall'equazione (2.19), mentre $\boldsymbol{\varphi}$ nell'equazione (2.21) è ottenuto dall'equazione (2.20). Si può ora scrivere la variazione prima del funzionale come:

$$\begin{aligned} \delta \mathcal{J} &= \oint_S \left(\frac{\partial j}{\partial f_i} \partial_n f_i + \frac{\partial j}{\partial T} \partial_n T + \frac{\partial j}{\partial (\partial_n \tilde{\nu})} \partial_n^2 \tilde{\nu} \right) \alpha \, ds \\ &- \oint_S \left(\frac{\partial j}{\partial \mathbf{n}} + \frac{\partial j}{\partial \mathbf{j}} p - \frac{\partial j}{\partial \underline{\boldsymbol{\sigma}}} \right) \cdot \nabla_S \alpha \, ds \\ &- \oint_S (\hat{g} + 2H_m j) \alpha \, ds - 2 \oint_S \Psi_\nu g_4 (p \mathbf{n} - \mathbf{n} \cdot \underline{\boldsymbol{\sigma}}) \cdot \nabla_S \alpha. \end{aligned} \quad (2.23)$$

È importante notare che nè nelle equazioni di Navier-Stokes aggiunte (2.16), nè nelle equazioni di Spalart-Allmaras aggiunte (2.17) è presente la variabile aggiunta Ψ_d ; inoltre neanche nella variazione del funzionale (2.23) è presente tale variabile, di conseguenza non è necessario risolvere l'equazione aggiunta (2.18). per il calcolo della sensitività.

2.1.1 Ottimizzazione della forza aerodinamica

Considerando ora funzionali che dipendono solo da \mathbf{f} nella forma seguente:

$$j(\mathbf{f}) = \mathbf{f} \cdot \mathbf{d}$$

dove \mathbf{d} è un vettore costante, si può giungere ad una formula semplificata della variazione di J ; questo è il tipico caso di ottimizzazione di forza aerodinamica generata da una superficie. In questo caso le condizioni al contorno delle equazioni aggiunte si riducono a:

$$\Psi_{\underline{v}} = \Psi_d = 0, \quad \varphi = \mathbf{d}, \quad \partial_n \Psi_5 = \frac{\mathbf{g}_1 \cdot \mathbf{d}}{g_2} \quad (2.24)$$

e la variazione del funzionale si può scrivere come:

$$\begin{aligned} \delta \mathcal{J} &= \oint_S \mathbf{d} \cdot \partial_n \mathbf{f} \alpha \, ds - \oint_S (p \mathbf{d} - \mathbf{d} \cdot \underline{\underline{\sigma}}) \cdot \nabla_S \alpha \, ds \\ &\quad - \oint_S (\hat{g} + 2H_m j) \alpha \, ds. \end{aligned}$$

Da cui integrando per parti si ottiene:

$$\begin{aligned} \delta \mathcal{J} &= \oint_S \partial_n (p \mathbf{d} \cdot \mathbf{n} - \mathbf{d} \cdot \underline{\underline{\sigma}} \cdot \mathbf{n}) \alpha \, ds + \oint_S \nabla_S \cdot (p \mathbf{d} - \mathbf{d} \cdot \underline{\underline{\sigma}}) \alpha \, ds \\ &\quad - \oint_S (\hat{g} + 2H_m j) \alpha \, ds = \oint_S \nabla \cdot (p \mathbf{d} - \mathbf{d} \cdot \underline{\underline{\sigma}}) \alpha \, ds \\ &\quad - \oint_S \hat{g} \alpha \, ds = - \oint_S \hat{g} \alpha \, ds \end{aligned}$$

dove è stata sfruttata la relazione:

$$\nabla \cdot \mathbf{q} = \partial_n (\mathbf{q} \cdot \mathbf{n}) + \nabla_S \cdot \mathbf{q} - 2H_m \mathbf{q} \cdot \mathbf{n},$$

valida per un generico vettore \mathbf{q} in un sistema di coordinate di S . È stata inoltre sfruttata l'identità:

$$\nabla \cdot (p \mathbf{d} - \mathbf{d} \cdot \underline{\underline{\sigma}}) = (\nabla p - \nabla \cdot \underline{\underline{\sigma}}) \cdot \mathbf{d} = 0 \quad \text{su } S$$

ottenuta dall'equazione della quantità di moto sulla superficie S .

L'espressione finale della variazione prima del funzionale dipende dalla funzione \hat{g} , che può essere riscritta come segue [4]:

$$\delta \mathcal{J} = - \oint_S \hat{g} \alpha \, ds = \oint_S \left(\mathbf{n} \cdot \underline{\underline{\Sigma}}^\varphi \cdot \partial_n \mathbf{v} - \mu_{\text{tot}}^2 C_p \nabla_S \Psi_5 \cdot \nabla_S T \right) \alpha \, ds \quad (2.25)$$

dove $\underline{\underline{\Sigma}}^\varphi$ dipende dai gradienti delle variabili aggiunte:

$$\underline{\underline{\Sigma}}^\varphi = \mu_{\text{tot}}^1 \left(\nabla \varphi + (\nabla \varphi)^T - \frac{2}{3} \underline{\underline{I}} \nabla \cdot \varphi \right). \quad (2.26)$$

La scrittura della variazione del funzionale aumentato è ora completa. Come si può notare dall'equazione (2.25), il parametro di deformazione normale α è

stato isolato, quindi \hat{g} è essenzialmente un gradiente locale, che chiameremo *sensitività aerodinamica*. Esso esprime, punto per punto sulla superficie S , l'ampiezza della deformazione normale che produce la variazione maggiore della funzione costo. Si evidenzia inoltre che nell'espressione della sensitività non è più presente il tensore degli sforzi, questo implica che non è possibile richiedere la minimizzazione della forza viscosa generata dalla superficie per qualsiasi funzione costo che implichi solo la forza aerodinamica totale.

2.1.2 Ipotesi di frozen viscosity

Tale ipotesi prevede che la variazione della viscosità rispetto alle variabili di progetto può essere trascurata se confrontata con gli altri termini delle equazioni aggiunte. Quindi l'equazione della turbolenza è utilizzata per il calcolo delle variabili del flusso, ma non viene considerata per il procedimento di scrittura delle equazioni aggiunte. Dunque vengono solo utilizzate le equazioni aggiunte di Navier-Stokes:

$$A_U^U \Psi_U = 0$$

con le seguenti condizioni al contorno:

$$\begin{aligned} \varphi_i &= \frac{\partial j}{\partial f_i} \quad \text{su } S \\ \partial_n \Psi_5 &= \frac{1}{g_2} \left(\frac{j}{T} + \mathbf{g}_1 \cdot \boldsymbol{\varphi} \right) \quad \text{su } S. \end{aligned}$$

Nella Figura 2.2 relativa al riferimento [4] si evidenzia come questa ipotesi renda imprecisa la sensitività calcolata; infatti il caso analizzato è il flusso transonico intorno ad un profilo RAE-2822. Si può vedere come al bordo d'uscita il segno della sensitività sia opposto al riferimento ottenuto tramite differenze finite.

In pratica ciò che interessa maggiormente è la *forma* della sensitività poiché l'entità della deformazione può essere modificata con opportune scalature. Per quanto riguarda l'inversione del segno al bordo d'uscita, si può vedere in Figura 2.3 che questo crea una piccola inefficienza nell'ottimizzazione nel senso che non raggiunge il punto di minimo della funzione obiettivo. Discussioni sui punti più sensibili della superficie da ottimizzare saranno fatte nei prossimi capitoli.

Purtroppo nel codice il calcolo aggiunto è stato implementato solo con questa ipotesi, quindi sarà utilizzata questa ipotesi nei calcoli turbolenti.

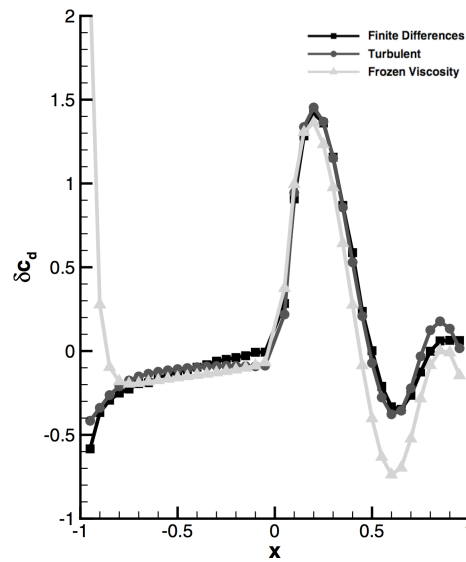


Figura 2.2: Confronto tra la sensitività calcolata tramite ipotesi di Frozen Viscosity, differenze finite e calcolo aggiunto completamente turbolento [4]; i valori negativi delle ascisse rappresentano il ventre del profilo, quelli positivi il dorso.

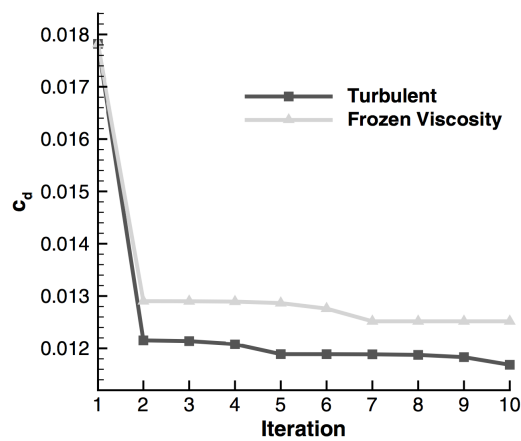


Figura 2.3: Confronto tra l'ottimizzazione con l'ipotesi di Frozen Viscosity e caso totalmente turbolento [4].

2.2 Definizione delle variabili di progetto

È stato appena mostrato che, utilizzando il metodo aggiunto continuo è possibile calcolare la variazione di una funzione costo rispetto alla deformazione normale infinitesima della superficie che si vuole ottimizzare. La scelta più intuitiva sembrerebbe quella di utilizzare come variabili di progetto tutti i nodi della mesh che appartengono alla superficie da modificare. Solitamente nella pratica non si segue questo approccio, bensì si ricorre ad un set ridotto di variabili (possibilmente una base completa) sul quale proiettare la sensitività geometrica. A questo punto la superficie viene modificata e la mesh riadattata alla nuova geometria; questo permette di avere una superficie modificata più regolare.

Di seguito si riportano due diverse scelte delle variabili di progetto utilizzate nei risultati numerici di questa tesi; per il caso bidimensionale sono state utilizzate funzioni di forma di Hicks-Henne, mentre per il caso tridimensionale è stato usato il metodo FFD (*Free Form Deformation*).

2.2.1 Funzioni di forma di Hicks-Henne

Sono funzioni (*bump functions*) che vengono sommate alla forma iniziale del profilo con lo scopo di modificarne la geometria:

$$y_{\text{new}} = y_{\text{old}} + \sum_{n=1}^N \delta_n f_n(x),$$

dove N è il numero di variabili di design, e δ_n è la differenza tra l'ampiezza precedente e quella successiva della variabile di design considerata, ovvero la deformazione da imporre.

Le funzioni di Hicks-Henne, con il massimo collocato nel punto x_n , sono definite come:

$$f_n(x) = \sin^3(\pi x^{e_n}), \quad e_n = \frac{\log(0.5)}{\log(x_n)}, \quad x \in [0, 1].$$

Queste funzioni vengono applicate separatamente al dorso e al ventre del profilo per ottenere la nuova forma, poi si procederà alla deformazione della mesh. Nell'immagine di seguito sono proposte le funzioni con ampiezza δ_n pari a 0.01 e nove variabili di design (x_n) equispaziate da 0.1 a 0.9:

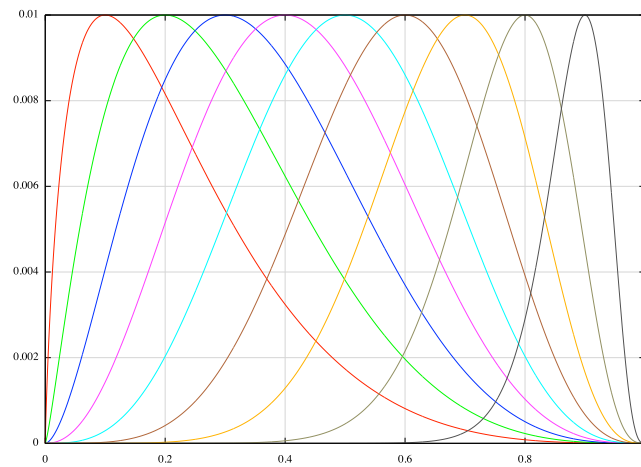


Figura 2.4: Funzioni di Hicks-Henne.

La variazione dei punti del profilo è data dalla somma, punto per punto, di tutte le funzioni proposte precedentemente, come rappresentato in Figura 2.5. Grazie all'utilizzo di queste funzioni è possibile ottenere una deformazione regolare della forma iniziale, nella Figura è stata rappresentata solo la deformazione.

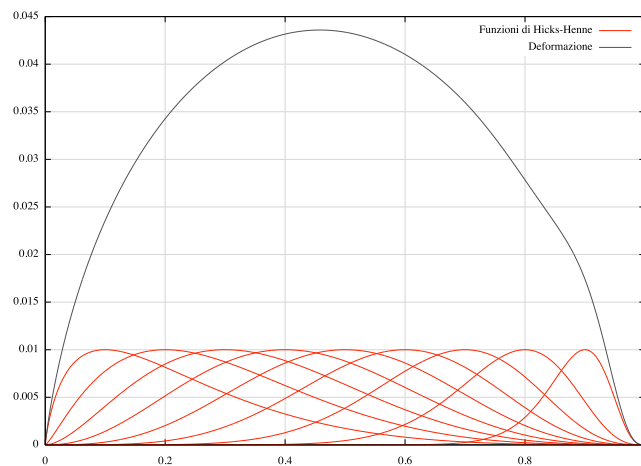


Figura 2.5: Deformazione tramite funzioni di Hicks-Henne.

2.2.2 Metodo FFD

Come già accennato in precedenza tale metodo è utilizzato in casi tridimensionali per deformare superfici o volumi. È un metodo molto versatile che consente, tra le altre cose, di essere applicato localmente, garantendo la continuità con le regioni adiacenti indeformate; inoltre è possibile imporre dei vincoli geometrici, ad esempio la conservazione del volume.

Come esposto nell'articolo di Sederberg et al. [18], una buona analogia fisica per la FFD è considerare un parallelepipedo di plastica flessibile e trasparente, nel quale sono inseriti diversi oggetti, come mostrato in Figura 2.6.

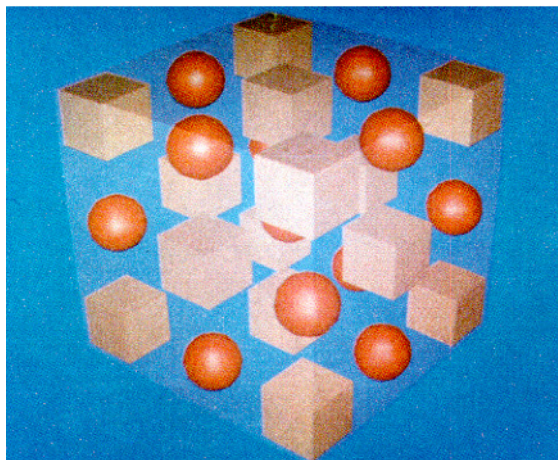


Figura 2.6: Parallelepipedo indeformato [18].

Gli oggetti sono da considerarsi flessibili, in modo che si possano deformare come le pareti plastiche del parallelepipedo. In Figura 2.7 il parallelepipedo è stato deformato così come gli oggetti all'interno e si può notare che la deformazione degli oggetti è consistente con quella del parallelepipedo.

Matematicamente il metodo FFD è definito come un prodotto tensoriale tri-variato di polinomi di Bernstein; tali polinomi sono una combinazione lineare di polinomi base di Bernstein definiti come:

$$B_i^j(x) = \binom{j}{i} x^i (1-x)^{j-i}, \quad i = 0, \dots, j$$

i coefficienti della combinazione lineare P_{ijk} sono, dal punto di vista geometrico, i punti di controllo del metodo, allora la deformazione è:

$$X_{\text{FFD}}(u, v, w) = \sum_{i,j,k=0}^{l,m,n} P_{ijk} B_i^l(u) B_j^m(v) B_k^n(w)$$

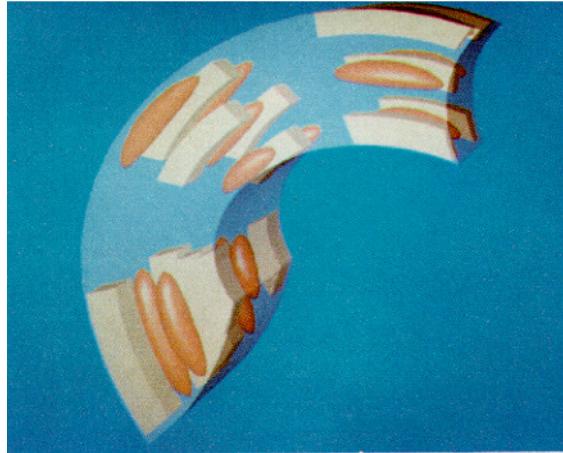


Figura 2.7: Parallelepipedo deformato [18].

dove $u, v, w \in [0, 1]$ e l, m, n rappresentano i gradi dei polinomi base. Come si può notare dalla Figura 2.8, esiste una relazione chiara tra la deformazione del solido e i punti di controllo. In questo caso i punti di controllo diventano

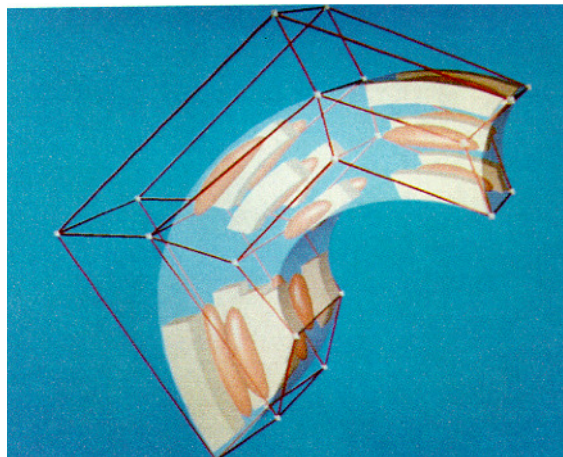


Figura 2.8: Parallelepipedo deformato e punti di controllo [18].

le variabili di design poichè possono deformare la forma del solido. Una volta che i punti di controllo sono stati spostati per ottenere la deformazione voluta, si applica la relazione precedente per ottenere le nuove coordinate cartesiane del solido.

2.3 Algoritmo di ottimizzazione vincolata

Risolvere un problema di ottimizzazione vuol dire trovare i punti di minimo (o massimo) locale (o globale) del seguente problema:

$$\min_{x \in \mathcal{X}} f(x) \quad (2.27)$$

dove $f(x)$ rappresenta la funzione obiettivo e $\mathcal{X} \subseteq \mathbb{R}^n$ è l'insieme che definisce i vincoli di uguaglianza e disuguaglianza:

$$\mathcal{X} : \begin{cases} h_j(x) = 0, & j = 1, \dots, p \\ g_i(x) \leq 0, & i = 1, \dots, m. \end{cases} \quad (2.28)$$

La struttura del problema di ottimizzazione (2.27) è molto generale; per categorie di problemi più specifici esistono algoritmi che sono particolarmente efficienti. Per casi più generali, invece, esistono diversi approcci, tra cui una famiglia di metodi che risolve una *sequenza di sottoproblemi vincolati* più semplici del problema (2.27). Tra questi, i metodi SQP (*Sequential Quadratic Programming*) costituiscono un'estensione del metodo di Newton ai casi di ottimizzazione vincolata.

2.3.1 Metodo risolutivo

Così come per i metodi non vincolati, anche per i metodi SQP è necessario introdurre strategie risolutive che garantiscano la convergenza globale del metodo, ovvero che la convergenza al punto di minimo x^* sia garantita a partire da qualsiasi punto iniziale $x_1 \in \mathcal{X}$. Si utilizza a tal proposito un metodo iterativo che calcola, ad ogni passo k la direzione d_k di discesa al minimo della funzione obiettivo e aggiornando le variabili tramite l'equazione:

$$x_{k+1} = x_k + d_k. \quad (2.29)$$

La ricerca della direzione di discesa al minimo viene effettuata risolvendo un sottoproblema di programmazione quadratica, ottenuto minimizzando lo sviluppo al second'ordine del Lagrangiano del problema di ottimizzazione non lineare (2.27) (2.28):

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{j=1}^p \lambda_j g_j(x) - \sum_{i=1}^m \sigma_i h_i(x)$$

dove λ_j e σ_i sono i moltiplicatori di Lagrange collegati rispettivamente ai vincoli di disuguaglianza e uguaglianza. Definendo i Jacobiani dei vincoli

come:

$$\begin{aligned} \left[\nabla h_1(x) \quad \cdots \quad \nabla h_p(x) \right]^T &= A(x) \\ \left[\nabla g_1(x) \quad \cdots \quad \nabla g_m(x) \right]^T &= B(x), \end{aligned}$$

si può scrivere il problema classico di programmazione quadratica (QP) ottenuto dalla linearizzazione al second'ordine del Lagrangiano:

$$\begin{aligned} \min_d \quad & \frac{1}{2} d_k^T \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k, \sigma_k) d_k + \nabla f(x_k)^T d_k \\ & A(x_k) d_k + h(x_k) = 0 \\ & B(x_k) d_k + g(x_k) \leq 0, \end{aligned} \tag{2.30}$$

dove la funzione obiettivo è lo sviluppo al secondo ordine di $f(x)$ in x_k , avendo sostituito il termine di ordine maggiore con un termine quadratico che contiene il Lagrangiano e i vincoli sono dati dallo sviluppo al primo ordine delle funzioni $h(x)$ e $g(x)$. Il calcolo dell'Hessiano della Lagrangiana ($\nabla_{xx}^2 \mathcal{L}$) è un procedimento oneroso, motivo per il quale esso viene calcolato a partire dalle derivate prime note ad ogni passo iterativo; gli algoritmi di ottimizzazione che sfruttano questa procedura sono detti *quasi Newtoniani*.

2.3.2 I metodi Sequential Quadratic Programming

Tali metodi furono introdotti per la prima volta da Wilson [22] e successivamente studiati negli anni '70. Di seguito vengono riportate le caratteristiche di questi metodi, di cui fa parte il metodo SLSQP (*Sequential Least Squares Quadratic Programming*) utilizzato in SU^2 per il processo di ottimizzazione geometrica.

Questi metodi, come già accennato in precedenza, hanno la particolarità di risolvere il problema (2.27) ricercando la soluzione attraverso una sequenza di sottoproblemi vincolati più semplici (2.30). L'insieme delle condizioni necessarie per la soluzione dei problemi di programmazione non lineare, con vincoli che soddisfano la cosiddetta condizione di qualificazione dei vincoli, prende il nome di condizione di Karush-Kuhn-Tucker (KKT):

$$\begin{cases} \nabla_x \mathcal{L}(x^*, \lambda^*, \sigma^*) = 0 \\ h_j(x^*) = 0, & j = 1, \dots, p, \\ g_i(x^*) \leq 0, & i = 1, \dots, m, \\ \sigma_i^* g_i^*(x^*) = 0, & i = 1, \dots, m, \\ \sigma^* \geq 0 \end{cases} \tag{2.31}$$

dove x^* è un punto di minimo locale per $f(x)$ in \mathcal{X} in cui è soddisfatta la condizione di qualificazione dei vincoli, λ^* e σ^* sono i moltiplicatori di Lagrange. Il vettore $(x^*, \lambda^*, \sigma^*)$ è un *punto di Karush-Kuhn-Tucker*.

Senza perdita di generalità, si considera il caso di soli vincoli di uguaglianza, mentre si rimanda a [12] per il caso con vincoli di disuguaglianza.

La condizione (2.31) sarà quindi:

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ h(x) \end{bmatrix} = \begin{bmatrix} \nabla_x \mathcal{L}(x, \lambda) \\ h(x) \end{bmatrix} = 0,$$

che rappresenta un sistema non lineare la cui soluzione si può ottenere applicando il metodo di Newton:

$$F(x_k + \Delta x_k, \lambda_k + \Delta \lambda_k) \approx F(x_k, \lambda_k) + J(x_k, \lambda_k) \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = 0$$

ovvero:

$$\underbrace{\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k) & -A(x_k)^T \\ A(x_k) & 0 \end{bmatrix}}_{J(x_k, \lambda_k)} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = - \underbrace{\begin{bmatrix} \nabla_x \mathcal{L}(x_k, \lambda_k) \\ h(x_k) \end{bmatrix}}_{F(x_k, \lambda_k)}. \quad (2.32)$$

Quindi all'iterazione successiva è possibile aggiornare le variabili come segue:

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix}.$$

Attraverso opportuni passaggi esposti nel riferimento [12], l'equazione (2.32) può essere riscritta nel modo seguente:

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k) & -A(x_k)^T \\ A(x_k) & 0 \end{bmatrix} \begin{pmatrix} d_k \\ \lambda_{k+1} \end{pmatrix} = - \begin{bmatrix} \nabla f(x_k) \\ h(x_k) \end{bmatrix} \quad (2.33)$$

e la variabile indipendente al passo $k+1$ può essere aggiornata con l'equazione (2.29) essendo d_k la soluzione del sistema lineare scritto precedentemente. Si può dimostrare [8] che il sistema (2.33) è equivalente al sistema che si ottiene applicando la condizione di Karush-Kuhn-Tucker (2.31) al QP (2.30). È quindi evidente che si sfrutta tale equivalenza per risolvere il problema più semplice di programmazione quadratica invece del problema iniziale di ottimizzazione vincolata. I vari metodi della classe degli SQP si distinguono per il modo in cui valutano l'Hessiano e il metodo di risoluzione utilizzato per calcolare la soluzione del sistema lineare (2.33).

Nel software SU^2 si utilizza un particolare metodo della classe degli SQP, ovvero l'SLSQP (*Sequential Least Squares Quadratic Programming*); tale metodo utilizza una forma modificata del metodo BFGS (*Broyden-Fletcher-Goldfarb-Shanno*) per il calcolo dell'Hessiano del Lagrangiano. È un metodo

utilizzato per la risoluzione di problemi di ottimizzazione non lineare non vincolata che è stato riadattato per valutare l'Hessiano ad ogni passo dell'SL-SQP. Per quanto riguarda la soluzione del sistema lineare (2.33), si utilizza il metodo ai minimi quadrati con vincolo di non negatività sulle variabili NNLS (*Non-Negative Least Squares*) introdotto da Lawson e Hanson [13].

Capitolo 3

Il codice SU^2

In questo capitolo si esporranno la struttura e i moduli principali del codice *Stanford University Unstructured* (SU^2) [14]. Il codice è stato sviluppato dal Dipartimento di Aeronautica e Astronautica dell'Università di Stanford e rappresenta un programma innovativo nell'ambito della CFD. Infatti è una *software suite* open-source scritta in C++ per la risoluzione di problemi descritti da equazioni alle derivate parziali (PDE), inoltre presenta diversi strumenti (scritti in linguaggio Python) che permettono di eseguire, tra le altre cose, ottimizzazione geometrica o calcolo parallelo. Nonostante il programma sia stato sviluppato per applicazioni di CFD e ottimizzazione geometrica, può essere esteso a qualsiasi problema con differenti equazioni di governo.

Il fatto stesso di essere stato concepito come un codice versatile e facilmente generalizzabile, ha obbligato gli sviluppatori a progettare il codice in moduli separati, che rappresentano modelli fisici e procedure differenti. Sempre basandosi su questa filosofia, è stato possibile sviluppare sia metodi a volumi finiti (FVM) che ad elementi finiti (FEM), sistemi aggiunti corrispondenti e solutori multi-fisici capaci di combinare entrambi gli approcci. L'utilizzo di *classi* permette di identificare chiaramente le parti di codice, in modo da poter facilmente modificarle o migliorarle lasciando inalterate le altre funzioni del software.

L'ultima versione disponibile sul sito <http://su2.stanford.edu> è la 2.0, ma vengono frequentemente rilasciate versioni beta per sviluppatori (attualmente l'ultima versione disponibile è la 2.0.5). Il programma include numerosi moduli per la risoluzione di sistemi di PDE su griglie non strutturate, che sono, tra gli altri:

- `SU2_CFD`: il modulo principale per la risoluzione delle equazioni.
- `SU2_DDC`: il modulo che suddivide il dominio per il calcolo in parallelo.

- SU2.GPC: permette il calcolo della sensitività aerodinamica sulla superficie da ottimizzare proiettando la sensitività su un set ridotto di variabili di design.
- SU2.MDC: è il modulo che deforma la mesh a seguito di una variazione di geometria.

Nella sezione successiva sarà descritta l'architettura del software e particolare attenzione verrà posta nella descrizione della struttura delle classi ad oggetti.

3.1 Struttura del codice

Si farà riferimento al modulo SU2.CFD, che è il modulo principale del programma, ma molte di queste classi sono utilizzate anche negli altri moduli. Nella Figura 3.1 è riportata la gerarchia delle classi di SU2.CFD che sono le seguenti:

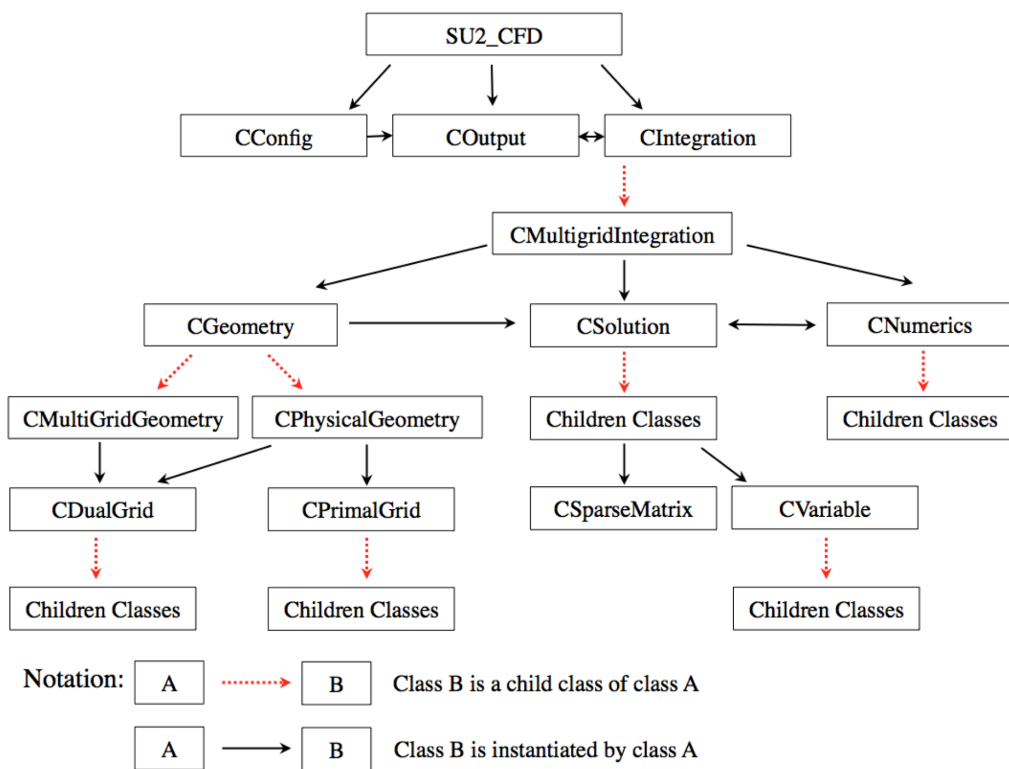


Figura 3.1: Gerarchia delle classi del modulo SU2.CFD

- **CConfig**: legge dal file di configurazione (estensione `.cfg`) la descrizione fisica, matematica e numerica del caso da simulare.
- **COutput**: scrive l'output della simulazione in un formato che viene specificato dall'utente all'interno del file di configurazione (Paraview, Tecplot, o `.csv`).
- **CIntegration**: risolve le equazioni di governo richiamando la classe figlia **CMultiGridIntegration**. Questa classe è richiamata sia se la simulazione viene effettuata su griglia singola che multipla, e connette le seguenti sottoclassi: **CGeometry**, **CSolution** e **CNumerics** per effettuare l'integrazione in spazio e tempo.

La numerica del codice è tutta concentrata nelle sottoclassi di **CMultiGridIntegration** che verranno trattate di seguito.

Classe **CGeometry**

Questa classe legge e processa il file della mesh (in formato nativo `.su2`) e include diverse classi tra cui:

- **CPhysicalGeometry**: costruisce la mesh duale, da quella in input, sulla quale si basa la formulazione a volumi finiti.
- **CMultiGridGeometry**: crea griglie consecutive sempre meno fitte a partire da quella originale, utilizzate per il multi-grid.
- **CPrimalGrid** e **CDualGrid**: sono le classi che definiscono le caratteristiche geometriche della mesh primaria e duale.

Classe **CSolution**

In questa classe sono definite le procedure per il calcolo della soluzione; ogni classe figlia rappresenta un solutore per le diverse equazioni di governo, come mostrato in Figura 3.2. Ad esempio **CEulerSolution** è il solutore delle equazioni di Eulero e **CAdjEulerSolution** è il solutore delle equazioni aggiunte di Eulero. Queste sottoclassi ne richiamano diverse altre in **CNumerics** per discretizzare i vari termini delle equazioni di governo. In generale le sottoclassi di **CSolution** hanno come classi istanziate:

- **CVariable**: è utilizzata per raccogliere le variabili, che dipendono dal tipo di equazioni, in ogni punto della griglia. Come mostrato in Figura 3.2 una classe figlia può essere ad esempio **CNSVariable** che raccoglie le variabili del sistema di equazioni di Navier-Stokes.

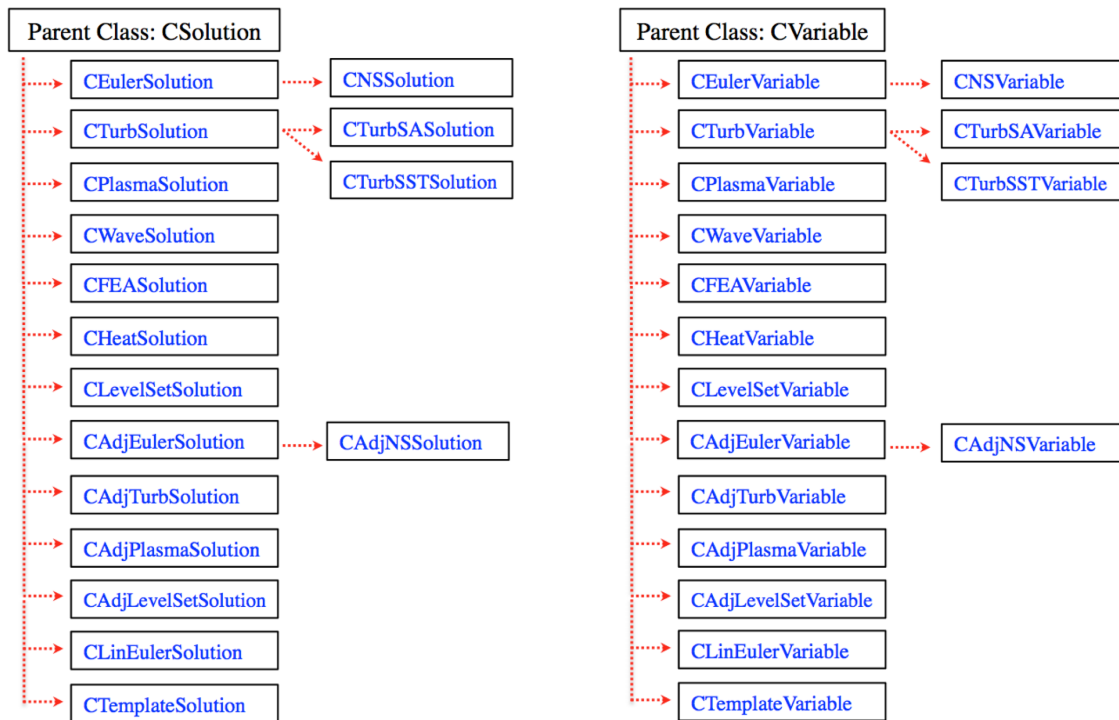


Figura 3.2: Classi figlie di CSolution e CVariable

- CSparseMatrix: raccoglie i Jacobiani dei flussi e i termini sorgente in matrici sparse per il calcolo implicito. Include anche i diversi metodi per la risoluzione di sistemi di equazioni lineari (LU-SGS, BiCGSTAB, GMRES).

Classe CNumerics

È la classe che discretizza i termini delle equazioni di governo utilizzando gli schemi numerici specificati nel file di configurazione. Ad esempio, in un calcolo implicito, questa classe calcola i residui e i Jacobiani in ogni nodo della mesh, utilizzando le variabili presenti in CVariables, passa i risultati a CSolution che a sua volta chiama le funzioni in CSparseMatrix per la risoluzione del sistema lineare. In Figura 3.3 sono elencate tutte le classi figlie di CNumerics.

Si riporta in Figura 3.4 la struttura per una simulazione RANS con integrazione esplicita nel tempo.

Il codice è stato concepito in modo da rimanere il più generale possibile e può approcciare senza particolari problemi di programmazione casi multidi-

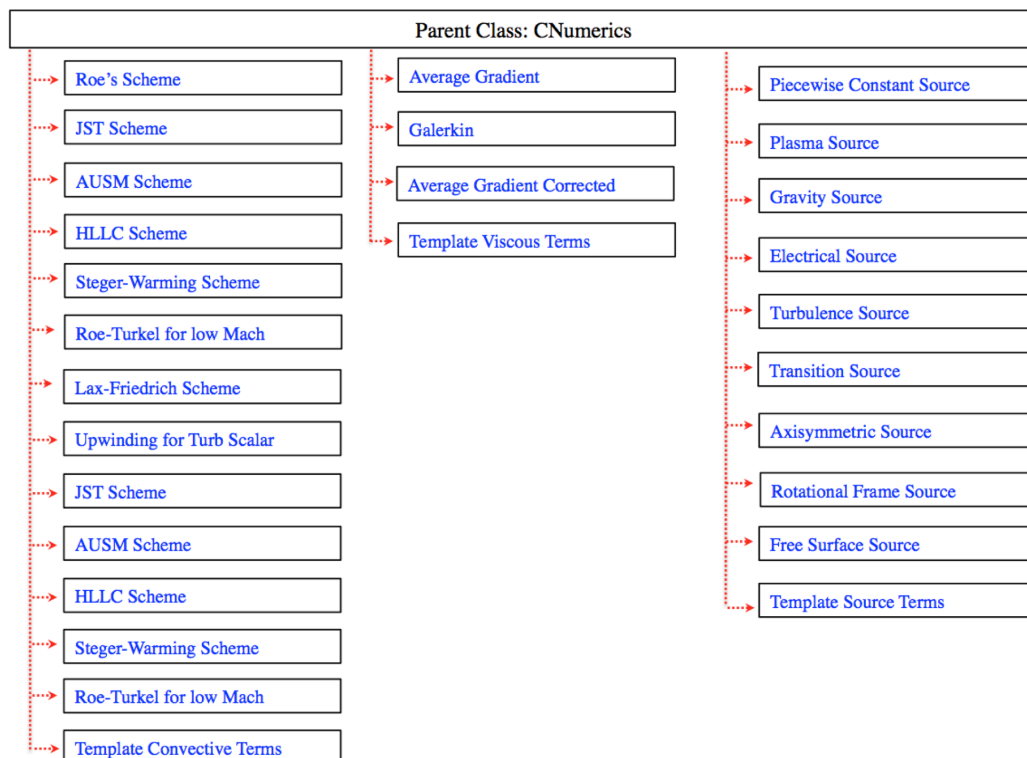


Figura 3.3: Classi figlie di CNumerics

sciplinari; di conseguenza, come è chiaro dalla Figura 3.4, le equazioni di Navier-Stokes e l'equazione della turbolenza vengono risolte separatamente. Per questo motivo sono necessarie due classi istanziate in `CSolution` (`CNSSolution` e `CTurbSolution`), così come due classi istanziate in `CVariable` (`CNSVariable` e `CTurbVariable`).

3.2 Formato nativo della mesh

Le mesh native hanno estensione `.su2` e sono in formato ASCII. Poiché il codice è scritto per lavorare con griglie non strutturate, sono richieste informazioni sui nodi e sulla connettività degli stessi, sul tipo di elemento tridimensionale, bidimensionale o lineare e infine da quali nodi esso è formato. Nella prima riga del file che definisce la mesh viene dichiarata la dimensione del problema, infatti SU^2 può operare sia su mesh bidimensionali che tridimensionali. Nella parte successiva del file sono descritti gli elementi interni

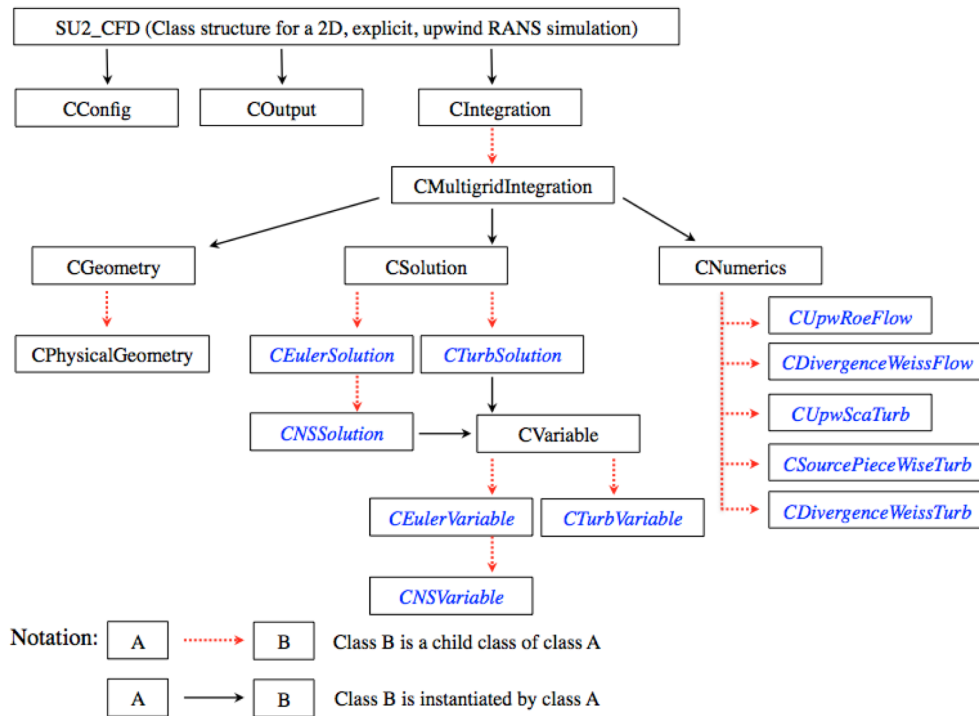


Figura 3.4: Struttura di una simulazione RANS con integrazione esplicita nel tempo.

nel modo seguente:

```

%
%Inner element connectivity
%
NELEM = 48970
9 0 1 3 2 0
9 1 4 5 3 1
9 4 6 7 5 2
    
```

dove ogni riga rappresenta la descrizione di un elemento interno. La prima colonna definisce l'identificatore del tipo di elemento che viene descritto nella riga (secondo il formato VTK). Seguono poi i nodi che compongono l'elemento descritto e il numero che identifica l'elemento costruito.

Una volta che la connettività è stata descritta, bisogna fornire le coordinate

di ogni nodo come esposto di seguito:

```

%
%Node coordinates
%
NPOIN = 49434
- 1.4008731282782294e - 01    - 1.6638704286217959e - 01    0
- 1.2947872467754354e - 01    - 1.5578240873270222e - 01    1
- 1.4164312923291425e - 01    - 1.6478390506414181e - 01    2
- 1.3097929203469980e - 01    - 1.5424567685235527e - 01    3
- 1.1887380719309064e - 01    - 1.4518144390092497e - 01    4
- 1.2031914461897376e - 01    - 1.4371109496123094e - 01    5
- 1.0839453032401439e - 01    - 1.3470607285589833e - 01    6
- 1.0978529214927452e - 01    - 1.3330132039213211e - 01    7

```

Dopo la dichiarazione del numero di nodi attraverso la definizione della variabili NPOIN, si passa alla definizione delle coordinate e ad ogni punto viene assegnato il numero che lo identifica.

L'ultima parte del file è costituita dalla definizione degli elementi dei contorni del dominio di calcolo:

```

NMARK = 2
MARKER_TAG = AIRFOIL
MARKER_ELEMS = 277
3    138    139
3    139    209
3    209    279
3    279    349
3    349    419
3    419    489

```

Si dichiara prima il numero di contorni nel dominio, che prendono il nome di *markers*; per ogni *marker* è necessario dichiarare il nome e il numero di elementi da cui è costituito. Ancora una volta gli elementi di ogni *marker* vengono dichiarati mediante un identificatore che dichiara il tipo di elemento e i punti da cui è costituito l'elemento di contorno.

3.3 Discretizzazione numerica

Ad oggi, la maggior parte dei metodi numerici per la risoluzione delle equazioni di governo della fluidodinamica utilizza una discretizzazione separata in spazio e tempo [3], tale tecnica prende il nome di *metodo delle linee* [17]. Questo metodo consente di selezionare schemi numerici di accuratezza differente per la discretizzazione spaziale e temporale.

Come già detto, il codice SU^2 è stato concepito per essere il più generale possibile e quindi per poter risolvere qualsiasi equazione alle derivate parziali che può essere discretizzata sia mediante metodo a volumi finiti che ad elementi finiti.

Nel seguito verrà esposto l'approccio a volumi finiti, utilizzato per la risoluzione delle equazioni della dinamica dei fluidi.

3.3.1 Integrazione spaziale

Il metodo a volumi finiti si basa sulle leggi di conservazione, rappresentate dal sistema di equazioni di Navier-Stokes (1.24). Il primo passo del metodo prevede la definizione di volumi di controllo, che non necessariamente coincidono con gli elementi della griglia. Si consideri quindi un generico volume di controllo che riterremo non variabile nel tempo; allora la derivata temporale delle variabili conservative può essere scritta nella forma:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega = \Omega \frac{\partial \mathbf{U}}{\partial t}$$

Gli integrali di superficie, che interessano i flussi convettivi e viscosi, vengono approssimati dalla somma dei flussi che attraversano le facce del volume di controllo. I flussi vengono considerati costanti sulla singola faccia del volume di controllo e valutati nel centro della stessa. Per quanto riguarda i termini sorgente, vengono considerati costanti nel volume di controllo, approssimazione accettabile finché tali termini non sono dominanti sugli altri.

Si consideri quindi un generico volume di controllo Ω_I a cui applicare le leggi di conservazione (1.24); a seguito della discretizzazione spaziale sopra descritta, queste ultime possono essere riscritte come:

$$\frac{\partial \mathbf{U}_I}{\partial t} = -\frac{1}{\Omega_I} \left[\sum_{m=1}^{N_F} \left[\tilde{\mathbf{F}}^c - \left(\mu^1 \tilde{\mathbf{F}}^{v1} + \mu^2 \tilde{\mathbf{F}}^{v2} \right) \right]_m \Delta S_m - (\mathbf{Q}\Omega)_I \right] \quad (3.1)$$

dove N_F rappresenta il numero delle facce del volume di controllo, ΔS_m è l'area della faccia m -esima del volume di controllo e $\tilde{\mathbf{F}}$ rappresenta la discretizzazione spaziale dei flussi. Il membro di destra dell'equazione (3.1)

prende il nome di *residuo*, quindi si può riscrivere l'equazione precedente nella seguente forma:

$$\frac{\partial \mathbf{U}_I}{\partial t} = -\frac{1}{\Omega_I} \mathbf{R}_I(\mathbf{U}). \quad (3.2)$$

È possibile scrivere la relazione per ogni volume di controllo nel dominio numerico per ottenere un sistema di equazioni differenziali ordinarie del primo ordine. Le equazioni sono di tipo iperbolico nel tempo, il che vuol dire che devono essere risolte avanzando con passi temporali partendo da una condizione iniziale

Finora si è solo parlato di volume di controllo senza specificare come esso è scelto all'interno del dominio. Esistono diverse metodologie per la definizione del volume di controllo, in SU^2 si utilizza l'approccio *vertex-centered* con griglia duale (Figura 3.5). In questo approccio le variabili del flusso si considerano nei vertici della griglia primaria, mentre i volumi di controllo sono costruiti unendo i centroidi degli elementi adiacenti, si crea così la cosiddetta griglia *duale*. Con questo approccio l'equazione (3.1) può essere riscritta

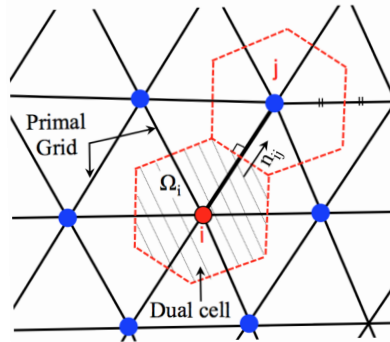


Figura 3.5: Griglia primaria e duale [14].

come segue:

$$\frac{\partial \mathbf{U}_i}{\partial t} = -\frac{1}{\Omega_i} \left[\sum_{j \in \mathcal{N}(i)} \left[\tilde{\mathbf{F}}_{ij}^c - \left(\mu^1 \tilde{\mathbf{F}}_{ij}^{v1} + \mu^2 \tilde{\mathbf{F}}_{ij}^{v2} \right) \right] \Delta S_{ij} - \mathbf{Q}_i \Omega_i \right] = -\frac{1}{\Omega_i} \mathbf{R}_i(\mathbf{U}) \quad (3.3)$$

dove $\mathcal{N}(i)$ rappresenta il numero di nodi adiacenti il nodo i -esimo.

È importante sottolineare che le variabili del flusso, sia quelle conservative sia quelle dipendenti, vengono immagazzinate negli stessi punti (in questo caso tutte nei vertici della griglia), per questo l'approccio è noto come schema *co-locato*; è una considerazione importante da fare perchè altri metodi, principalmente i metodi *pressure-based*, immagazzinano le variabili *pressure*

e velocità in punti diversi della griglia per sopprimere le oscillazioni spurie della soluzione.

Discretizzazione dei flussi convettivi

I flussi convettivi possono essere discretizzati utilizzando schemi centrali o upwind; nel codice sono implementati diversi schemi numerici tra cui JST, largamente utilizzato nelle simulazioni successive.

È uno schema centrale con dissipazione artificiale del terzo ordine, calcolata utilizzando le differenze dei Laplaciani completi (ordine maggiore) di due nodi adiacenti i, j e le differenze delle variabili conservative (ordine minore) negli stessi nodi. I due livelli di dissipazione sono combinati mediante l'utilizzo di rilevatori di variazione di pressione per utilizzare una minore dissipazione in vicinanza di onde d'urto. L'espressione del flusso convettivo numerico per lo schema JST è il seguente:

$$\tilde{\mathbf{F}}_{ij}^c = \tilde{\mathbf{F}}^c(\mathbf{U}_i, \mathbf{U}_j) = \mathbf{F}^c\left(\frac{\mathbf{U}_i + \mathbf{U}_j}{2}\right) \cdot \mathbf{n}_{ij} - \mathbf{d}_{ij}, \quad (3.4)$$

\mathbf{d}_{ij} rappresenta il termine di dissipazione artificiale ed ha la seguente espressione:

$$\mathbf{d}_{ij} = \left(\epsilon_{ij}^{(2)}(\mathbf{U}_j - \mathbf{U}_i) - \epsilon_{ij}^{(4)}(\nabla^2 \mathbf{U}_j - \nabla^2 \mathbf{U}_i)\right) \varphi_{ij} \lambda_{ij}, \quad (3.5)$$

dove $\nabla^2 \mathbf{U}_i$ rappresenta il Laplaciano completo, λ_{ij} rappresenta una stima del raggio spettrale del Jacobiano della faccia ij e φ_{ij} lo stretching nella mesh; questi elementi vengono calcolati nel seguente modo:

$$\begin{aligned} \nabla^2 \mathbf{U}_i &= \sum_{k \in \mathcal{N}(i)} (\mathbf{U}_k - \mathbf{U}_i), \\ \lambda_{ij} &= (|\mathbf{v}_i| + c_i + |\mathbf{v}_j| + c_j) \Delta S_{ij}, \quad \lambda_i = \sum_{k \in \mathcal{N}(i)} \lambda_{ik}, \\ \varphi_{ij} &= 4 \frac{\varphi_i \varphi_j}{\varphi_i + \varphi_j}, \quad \varphi_i = \left(\frac{\lambda_i}{4\lambda_{ij}}\right)^p, \\ \epsilon_{ij}^{(2)} &= \kappa^{(2)} s_2 \left(\frac{|\sum_{k \in \mathcal{N}(i)} (p_k - p_i)|}{\sum_{k \in \mathcal{N}(i)} (p_k + p_i)}\right), \quad \epsilon_{ij}^{(4)} = s_4 \max\left(0, \kappa^{(4)} - \epsilon_{ij}^{(2)}\right), \end{aligned}$$

dove p_i, \mathbf{u}_i, c_i rappresentano rispettivamente la pressione, la velocità del flusso e la velocità del suono nel nodo i , ΔS è l'area della faccia ij , s_2, s_4 sono parametri di stretching e κ_2, κ_4 sono parametri regolabili dall'utente.

Discretizzazione dei flussi viscosi

Per valutare i flussi viscosi si considerano le quantità del fluido e le loro derivate nel centro delle facce del volume di controllo. Le derivate possono

essere valutate sia con il metodo di Green-Gauss sia con il metodo dei minimi quadrati, poi vengono mediate per ottenere il valore sulla faccia della cella (media pesata). Per ridurre l'errore la media può essere corretta per ridurre l'errore dalla seguente formula:

$$\nabla\phi \cdot \mathbf{n}_{ij} = \frac{\phi_j - \phi_i}{|x_j - x_i|} \mathbf{s} \cdot \mathbf{n}_{ij} + \frac{1}{2} (\nabla\phi|_i + \nabla\phi|_j) \cdot (\mathbf{n} - (\mathbf{s} \cdot \mathbf{n}_{ij}) \mathbf{s}),$$

dove \mathbf{n}_{ij} è la normale alla faccia in comune tra la cella i e j , \mathbf{s} è il vettore che collega i centroidi di due celle adiacenti (media pesata corretta).

3.3.2 Integrazione temporale

Si passa ora alla discretizzazione temporale delle leggi di conservazione per avere un sistema discretizzato in spazio e tempo. Il programma utilizza la tecnica di *pseudo time-stepping* per ottenere la soluzione stazionaria. Questa tecnica può essere interpretata come un solutore iterativo per il problema stazionario associato e il passo temporale fittizio come il fattore di rilassamento. Poichè l'equazione (3.3) deve essere verificata in qualsiasi istante di tempo, si può decidere di valutare i residui $\mathbf{R}_i(\mathbf{U})$ al tempo t^n (metodi espliciti) o al tempo t^{n+1} (metodi impliciti). Nel codice sono implementate entrambi i metodi: multi-stage Runge-Kutta ed Eulero esplicito o Eulero implicito. Di seguito si riporta la trattazione per l'approccio implicito utilizzato nelle simulazioni successive.

Discretizzazione implicita

Come già detto, per ottenere una discretizzazione temporale implicita, i residui vanno valutati all'istante di tempo t^{n+1} :

$$\frac{\Omega_i}{\Delta t_i^n} \Delta \mathbf{U}_i^n = -\mathbf{R}_i(\mathbf{U}^{n+1}) \quad (3.6)$$

dove $\Delta \mathbf{U}_i^n = \mathbf{U}_i^{n+1} - \mathbf{U}_i^n$. I residui all'istante t^{n+1} sono incogniti, quindi è necessario linearizzare in t^n :

$$\begin{aligned} \mathbf{R}_i(\mathbf{U}^{n+1}) &= \mathbf{R}_i(\mathbf{U}^n) + \frac{\partial \mathbf{R}_i(\mathbf{U}^n)}{\partial t} \Delta t_i^n + \mathcal{O}(\delta t^2) = \\ &= \mathbf{R}_i(\mathbf{U}^n) + \sum_{j \in \mathcal{N}(i)} \frac{\partial \mathbf{R}_i(\mathbf{U}_i)}{\partial \mathbf{U}_j} \Delta \mathbf{U}_j^n + \mathcal{O}(\delta t^2). \end{aligned}$$

Trascurando i termini di secondo ordine, si può scrivere il seguente sistema lineare:

$$\left(\frac{\Omega_i}{\Delta t_i^n} \delta_{ij} + \frac{\partial \mathbf{R}_i(\mathbf{U}_i)}{\partial \mathbf{U}_j} \right) \Delta \mathbf{U}_j^n = -\mathbf{R}_i(\mathbf{U}^n). \quad (3.7)$$

La soluzione del sistema si ricava tramite i solutori lineari implementati in SU^2 : LU-SGS, GMRES, BiCGSTAB. Inoltre il codice utilizza la tecnica del local-time-stepping per accelerare la convergenza; tale tecnica permette che la soluzione avanzi, in ogni cella della mesh, con uno step temporale ottimale, che può essere differente da cella a cella.

3.4 Metodo Multigrid

Il metodo *multigrid* rappresenta una tecnica di accelerazione della convergenza implementata nel codice. Il concetto è quello di accelerare la convergenza della soluzione sulla griglia originaria, che chiameremo fine, applicando correzioni della stessa valutate su una griglia più lasca. La convergenza è accelerata perchè i metodi di risoluzione dei sistemi lineari smorzano efficacemente le oscillazioni dell'errore ad alta frequenza spaziale, ma non quelli a bassa frequenza, ovvero quelli che hanno una scala di lunghezza comparabile con il dominio di calcolo; ma utilizzando una griglia più lasca le oscillazioni di bassa frequenza, non smorzate, diventano di alta frequenza e possono essere smorzate dal solutore.

Si consideri quindi un problema non lineare del tipo:

$$L(u) = f, \quad \text{in } \Omega, \quad (3.8)$$

dove $L(u)$ è un operatore non lineare definito in Ω . A seguito di discretizzazione spaziale sulla griglia di calcolo (fine) si ottiene:

$$L_h(u_h) = f_h, \quad \text{in } \Omega_h \quad (3.9)$$

in questo caso $L_h(u_h)$ è l'operatore implicito discretizzato definito sulla griglia fine Ω_h . Il sistema può essere risolto con appositi solutori, ad esempio LU-SGS, per ottenere una soluzione approssimata \bar{u}_h , quindi l'equazione (3.9) può essere riscritta come segue:

$$L_h(\bar{u}_h) - f_h = r_h \quad (3.10)$$

dove r_h è per definizione il residuo. Sottraendo l'equazione (3.10) alla (3.9) si ottiene:

$$L_h(u_h) - L_h(\bar{u}_h) = -r_h.$$

Introducendo la correzione della soluzione c_h è possibile effettuare la sostituzione $u_h = \bar{u}_h + c_h$ ottenendo:

$$L_h(\bar{u}_h + c_h) - L_h(\bar{u}_h) = -r_h. \quad (3.11)$$

Ora l'obiettivo è quello di scrivere l'equazione (3.11) su una griglia più lasca di dimensioni caratteristiche H . Per fare ciò è necessario definire due operatori che interpolino il residuo (I_h^H) e la soluzione (\bar{I}_h^H) dalla griglia più fine a quella più lasca. Per riformulare l'equazione (3.11) sulla griglia più lasca basta sostituire L_h con L_H , \bar{u}_h e r_h con le loro interpolazioni $\bar{I}_h^H \bar{u}_h$ e $I_h^H r_h$ per ottenere l'equazione FAS (*Full Approximation Scheme*) del metodo:

$$L_H \left(\bar{I}_h^H \bar{u}_h + c_H \right) - L_H \left(\bar{I}_h^H \bar{u}_h \right) = -I_h^H r_h. \quad (3.12)$$

Sostituendo il residuo r_h con $L_h(\bar{u}_h) - f_h$ dalla (3.11) e definendo $\bar{u}_H = \bar{I}_h^H \bar{u}_h + c_H$ si può scrivere l'equazione sulla griglia più lasca:

$$L_H(\bar{u}_H) = L_H \left(\bar{I}_h^H \bar{u}_h \right) - I_h^H (L_h(\bar{u}_h) - f_h) = f_H + \tau_h^H, \quad \text{in } \Omega_H \quad (3.13)$$

dove il termine f_H è interpolato e non calcolato e la nuova variabile τ_h^H è definita come la correzione del residuo dalla griglia fine a quella lasca. Applicando ancora una volta un metodo di risoluzione iterativo, è possibile calcolare \bar{u}_H e aggiornare la soluzione sulla griglia più fine come segue:

$$\bar{u}_h^{\text{new}} = \bar{u}_h^{\text{old}} + I_H^h \left(\bar{u}_H^{\text{new}} - \bar{I}_H^h \bar{u}_h^{\text{old}} \right) \quad (3.14)$$

dove \bar{I}_H^h e I_H^h sono gli operatori di interpolazione dalla griglia più lasca a quella fine. Nella pratica questa tecnica è applicata in maniera riscorsiva su griglie via via più lasche per coprire la maggior parte dello spettro delle frequenze spaziali in modo da eliminare più velocemente le oscillazioni della soluzione. Ne consegue che l'errore della soluzione diminuirà a seguito di iterazioni di solutori lineari su griglie di dimensioni differenti; per abbattere il costo computazionale si cercherà inoltre di effettuare il maggior numero di iterazioni sulle griglie più lasche.

3.5 Deformazione della mesh

La deformazione della mesh a seguito di una deformazione di geometria è affidata al modulo `SU2_MDC`; quest'ultimo permette di modificare la griglia di calcolo attraverso differenti tecniche che possono essere selezionate nel file di configurazione richiesto in input dal codice. Tra le varie tecniche di deformazione della mesh nel codice SU^2 , il metodo che è stato più utilizzato in questa tesi è il metodo delle molle.

Tale metodo sfrutta l'analogia strutturale ed è possibile dimostrare che può essere implementato per evitare l'interpenetrazione degli elementi adiacenti. Si consideri per semplicità una mesh bidimensionale non strutturata a triangoli (il ragionamento può essere esteso al caso tridimensionale [6]); l'idea di

base del metodo è quella di interpretare la griglia di calcolo come un sistema pseudo-strutturale, continuo o discreto. Nel caso continuo, il comportamento quasi-statico del sistema meccanico, che chiameremo fittizio, è espresso dalla seguente equazione alle derivate parziali:

$$\operatorname{div}(E : \epsilon(u)) = 0$$

con la seguente condizione al contorno:

$$u = \bar{u} \quad \text{su } \Gamma_b,$$

dove u rappresenta lo spostamento della struttura fittizia, ϵ è il tensore degli sforzi, E è il tensore di elasticità, Γ_b è la superficie che deve essere deformata e \bar{u} rappresenta il campo di spostamenti imposti. La discretizzazione dell'equazione differenziale, se effettuata mediante metodo degli elementi finiti, porta al seguente sistema lineare:

$$K_{ii}u_i = -K_{ib}\bar{u}_b$$

dove K è la matrice di rigidezza del sistema fittizio, mentre \cdot_b rappresenta i nodi (gradi di libertà) della griglia su Γ_b mentre \cdot_i rappresenta i nodi interni della mesh. Di conseguenza gli spostamenti dei nodi interni della mesh dipendono da quelli dei nodi su Γ_b .

Un'alternativa a questa procedura è rappresentata dall'approccio discreto implementato in SU^2 , si interpreta la griglia direttamente come una struttura discreta, utilizzando molle concentrate. Ad esempio, concentrando tutta la deformabilità nei lati degli elementi della mesh, ed interpretandoli come molle lineari, si giunge al metodo delle molle lineari. In questo approccio un lato che connette due nodi i e j , di lunghezza l_{ij} , è caratterizzato da una rigidezza inversamente proporzionale alla sua lunghezza:

$$k_{ij} \propto \frac{1}{l_{ij}}.$$

Questa definizione della rigidezza è motivata dal fatto che se durante la deformazione della griglia i due nodi i e j si avvicinano, la lunghezza l_{ij} si riduce aumentando la rigidezza dell'elemento; questo provvede ad evitare che i due nodi si avvicinino troppo fino al limite a coincidere.

Questo metodo sembra funzionare bene finchè gli spostamenti imposti sono piccoli e la griglia di calcolo è abbastanza fine; in particolare, quando si richiedono spostamenti di ampiezza maggiore si incorre in una deformazione errata della mesh e gli elementi si incrociano. Questo è dovuto al fatto che, per come è stata modellata la rigidezza degli elementi, si evita che i nodi

collidano ma non è possibile evitare che un nodo attraversi un lato dell'elemento di cui fa parte. Una soluzione a tale inconveniente è l'utilizzo di molle torsionali concentrate nei nodi di ogni elemento di griglia, come proposto nei riferimenti [7, 6]. Si consideri ancora una volta una mesh non strutturata a triangoli, denotati con τ_{ijk} dove i, j, k sono i vertici, come mostrato in Figura 3.6 e A_{ijk} è l'area dell'elemento triangolare.

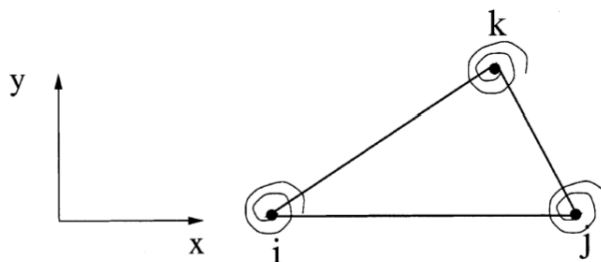


Figura 3.6: Schematizzazione del metodo a molle torsionali [6].

Definendo la rigidità della molla come segue [7]:

$$C_k^{ijk} = \frac{l_{ki}^2 l_{kj}^2}{4A_{ijk}^2}$$

si evita sia che due nodi collidano sia che un nodo possa attraversare il lato opposto dell'elemento a cui appartiene. Infatti, quando il vertice k si avvicina al lato $i-j$, l'area A_{ijk} si riduce, di conseguenza aumenta la rigidità della molla k che evita sia la sovrapposizione ad uno dei restanti vertici sia l'attraversamento del lato $i-j$. Per una trattazione più completa e rigorosa si fa ancora una volta riferimento a [7, 6].

3.6 Comandi per l'avvio del software

Il software, come già spiegato in questo Capitolo, è composto da diversi codici che svolgono compiti differenti e da programmi Python che eseguono procedure più complesse e automatizzate che richiedono l'utilizzo di più moduli di SU^2 .

Il software, per essere avviato, legge un file di configurazione di estensione `.cfg` in input in cui vengono impostate tutte le caratteristiche fisiche e matematiche del problema analizzato. Tramite il comando:

```
SU2_CFD configuration_file.cfg
```

si richiede che il software richiami l'eseguibile `SU2.CFD` per la risoluzione del problema, diretto o aggiunto, descritto nel file di configurazione.

Per impostare il calcolo parallelo si richiama il seguente script python:

```
parallel_computation.py -f configuration_file.cfg -p np
```

dove `np` rappresenta il numero di processori su cui si vuole suddividere il problema. Il programma precedente provvede a richiamare il codice per la decomposizione del dominio (`SU2.DDC`) per dividere la mesh di calcolo in `np` parti, poi richiama il codice per la risoluzione delle equazioni fluidodinamiche (`SU2.CFD`). Si noti che nel presente lavoro non è stato utilizzato il calcolo parallelo perchè si è rivelato essere poco affidabile dal punto di vista della stabilità numerica, infatti si sono riscontrati casi in cui, a seconda del numero di processori richiesti, la simulazione non sempre convergeva. Per questo motivo si è deciso di procedere sempre in modalità seriale.

Altra funzione python spesso utilizzata è quella relativa all'ottimizzazione geometrica:

```
shape_optimization.py -f configuration_file.cfg -p np
```

che automatizza il processo di ottimizzazione. Nel caso in cui sia specificato il numero di processori, richiama il codice `SU2.DDC` per la scomposizione del dominio, in caso contrario tutto il processo viene effettuato in seriale. Per prima cosa la funzione legge dal file di configurazione il problema di ottimizzazione in cui si specificano la funzione obiettivo (o le funzioni obiettivo nel caso in cui siano più di una), i vincoli (di uguaglianza o disuguaglianza) e il numero e il tipo di variabili di progetto (si veda la sezione 2.2 relativa alle variabili di progetto). La routine si basa sul metodo di ottimizzazione `SLSQP` descritto nella sezione 2.3.2, che richiede la valutazione della funzione obiettivo e delle sue derivate ad ogni iterazione del metodo, quindi effettua una simulazione delle equazioni dirette per valutare la funzione obiettivo. Dopo la prima simulazione, a seconda del numero di vincoli richiesti, vengono effettuate le simulazioni delle equazioni aggiunte per il calcolo dei gradienti della funzione obiettivo e dei vincoli. Una volta note funzione obiettivo e derivate, si può risolvere il problema di programmazione quadratica (QP) e trovare il vettore di discesa al minimo della funzione costo, che nel caso specifico rappresenta l'ampiezza della deformazione delle funzioni di Hicks-Henne (caso bidimensionale) o lo spostamento dei punti di controllo dell'FFD (caso tridimensionale). In base alla soluzione del problema QP, si aggiornano le variabili di progetto e si deforma la geometria della superficie e successivamente la griglia di calcolo; per fare ciò la routine python richiama il codice `SU2.MDC` che deforma la superficie e la griglia di calcolo utilizzando il metodo specificato nel file di configurazione; nelle simulazioni effettuate si è

utilizzato il metodo delle molle torsionali esposto nella sezione 3.5. Queste operazioni vengono effettuate finchè non si raggiunge un minimo locale, ovvero quando la derivata della funzione obiettivo rispetto alle variabili di design è al di sotto di una certa tolleranza o quando si raggiunge il numero massimo di iterazioni.

Capitolo 4

Risultati

Nel presente capitolo saranno presentati i risultati di alcune simulazioni effettuate con il codice SU^2 . L'obiettivo è quello di rendersi conto delle potenzialità e delle eventuali criticità del software, sia proponendo casi standard sia applicandolo a problemi diversi.

4.1 La mesh

Nel software non è stato implementato nessun programma che costruisca la mesh in formato nativo `.su2`, ma è possibile convertire una mesh da formato `.cgns` (*CFD General Notation System*) in formato nativo. Per questo motivo le mesh utilizzate per le simulazioni sono state costruite utilizzando il software ANSYS ICEM CFD[®], che può esportare in formato `.cgns` e quindi leggibile e convertibile da SU^2 . Si è riscontrato però un problema relativo all'orientazione degli elementi sulla superficie; infatti gli elementi di superficie creati da ANSYS ICEM CFD[®] non sono tutti orientati allo stesso modo, ed il convertitore in formato nativo non presenta alcun controllo sull'orientazione di tali elementi. Questo problema è rilevante quando si richiede la deformazione della geometria; infatti, poiché la deformazione ha direzione normale e verso stabilito dalla sensitività aerodinamica, è importante che tutti gli elementi abbiano la stessa orientazione in modo che le normali puntino tutte verso l'interno del profilo. Per questo motivo è stato necessario scrivere un semplice programma che riorientasse tutti gli elementi della superficie. Inoltre tutte le simulazioni sono state effettuate in modalità seriale poiché l'implementazione del calcolo parallelo si è rivelata poco affidabile nelle versioni di SU^2 utilizzate.

4.2 Simulazioni non viscosse

In questa prima sezione vengono proposti due casi, il primo bidimensionale e il secondo tridimensionale, di cui si mostra la soluzione diretta, quella aggiunta e l'ottimizzazione geometrica. Si tratta di variazioni limitate rispetto a casi descritti nella documentazione di SU^2 .

4.2.1 Caso 1: Profilo NACA 0012

Si simula il flusso non viscoso in regime transonico intorno al profilo NACA 0012 con corda unitaria. La mesh, bidimensionale, discretizza un dominio circolare, di raggio pari a 20 corde del profilo; la griglia è formata da 10216 elementi triangolari come mostrato in Figura 4.1.

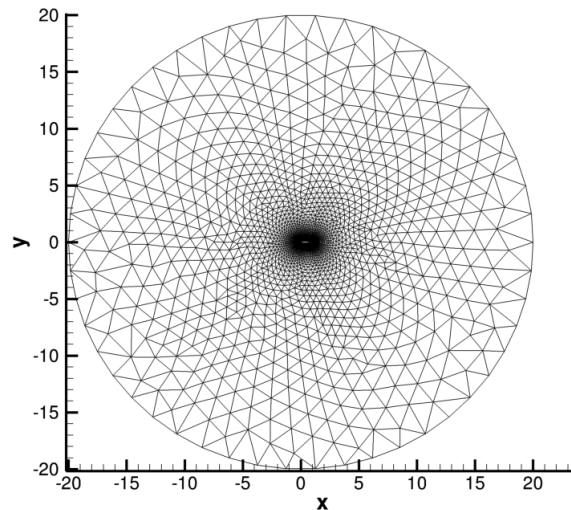


Figura 4.1: Dominio di calcolo.

Gli elementi in prossimità del bordo d'attacco e d'uscita sono di dimensioni ridotte, come mostrato in Figura 4.2, inoltre il profilo è discretizzato utilizzando 200 elementi lineari.

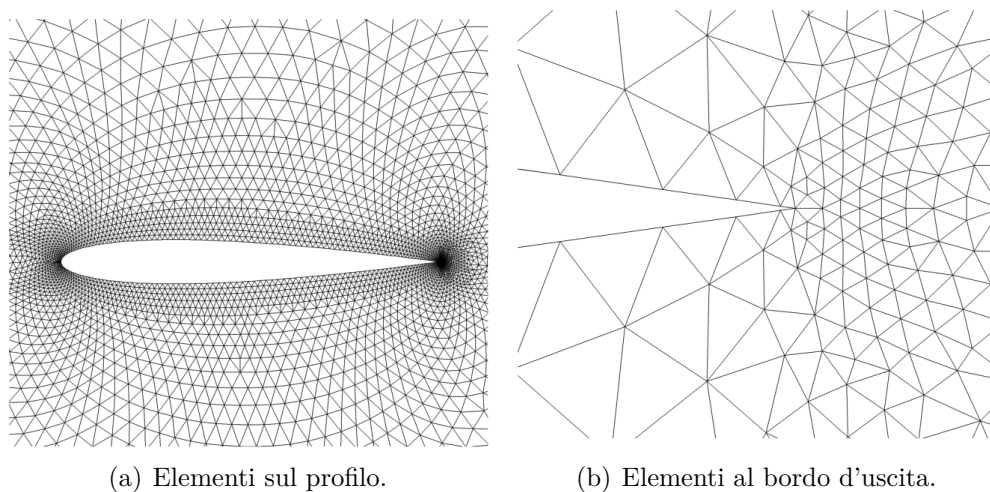


Figura 4.2: Griglia di calcolo intorno al NACA0012.

Calcolo diretto

I valori del flusso all'infinito che riproducono il caso di riferimento [20] sono i seguenti:

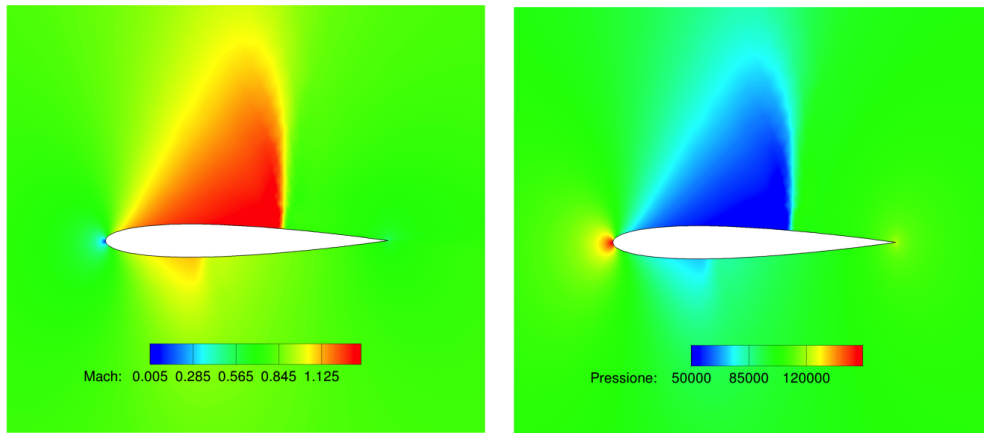
- $Mach = 0.8$;
- $\alpha = 1.25^\circ$;
- $P_\infty = 101\,325\text{ Pa}$;
- $T_\infty = 273.15\text{ K}$.

In Figura 4.3 si riportano il numero di Mach e il campo di pressione intorno al profilo; inoltre il coefficiente di pressione confrontato con i dati di riferimento è riportato in Figura 4.4.

Dalle Figure è evidente la presenza di un'onda d'urto sul dorso del profilo posta al 60% della corda.

I valori dei coefficienti aerodinamici calcolati sono:

- $C_L = 0.3269$;
- $C_D = 0.0213$.



(a) Numero di Mach

(b) Campo di pressione

Figura 4.3: Soluzione intorno al NACA 0012.

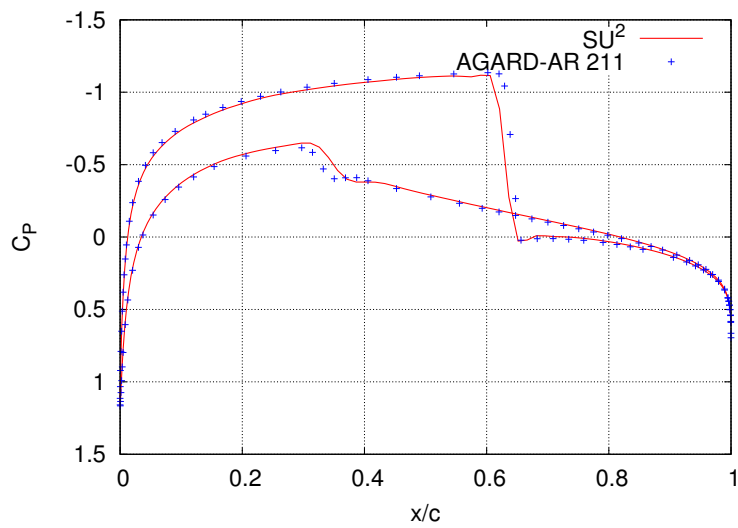


Figura 4.4: Coefficiente di pressione sul profilo e riferimento [20].

Il calcolo è stato effettuato su un portatile con processore Intel® Core i7 a 2.3 GHz e la soluzione è stata ottenuta in 27 s. In Figura 4.5 si riporta l'andamento del residuo dell'equazione di continuità.

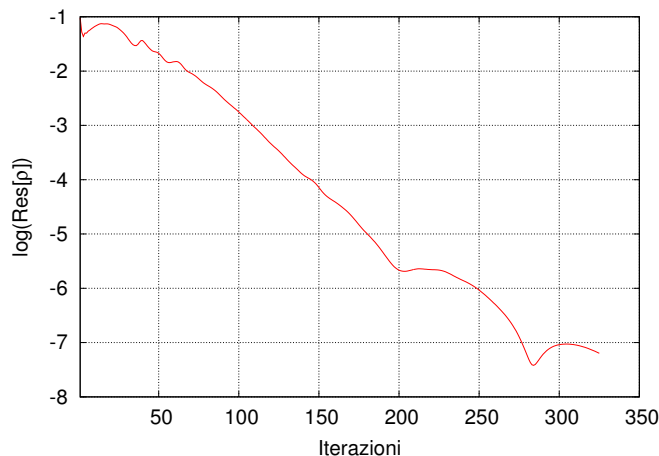


Figura 4.5: Residuo del calcolo diretto.

Calcolo aggiunto

Una volta ottenuta la soluzione diretta si può effettuare la simulazione aggiunta. Sono state effettuate due simulazioni aggiunte, la prima imponendo la resistenza aerodinamica come funzione obiettivo, nella seconda si è imposta la portanza.

Nella Figura 4.6 sono riportati gli andamenti delle sensitività sul profilo dove il semipiano delle ascisse negative rappresenta il ventre del profilo, l'altro semipiano il dorso.

Dalle figure precedenti si vede come i valori più elevati di sensitività sono concentrati al bordo d'uscita nonostante la presenza dell'onda d'urto. Escludendo tali punti dal grafico è possibile rendere più visibile l'andamento della sensitività sulle altre zone del profilo. Dalla Figura 4.7 si può osservare che la portanza presenta maggiore sensitività alle variabili di progetto intorno alla zona dell'onda d'urto, oltre a presentare altre due zone sensibili in prossimità del bordo d'attacco e in una zona sul ventre, tra il 20% e il 40%, in cui si ha una compressione abbastanza evidente. Per la resistenza, invece, la zona più sensibile è quella in prossimità del bordo d'attacco, in cui il valore della sensitività è più elevato rispetto alle altre zone (fatta eccezione ovviamente per il bordo d'uscita).

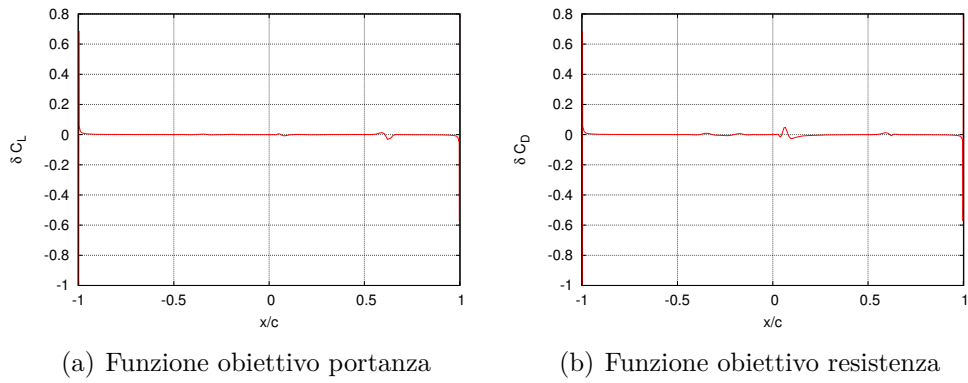


Figura 4.6: Sensitività aerodinamica normalizzata sul profilo; il semipiano negativo delle ascisse rappresenta il ventre, quello positivo il dorso.

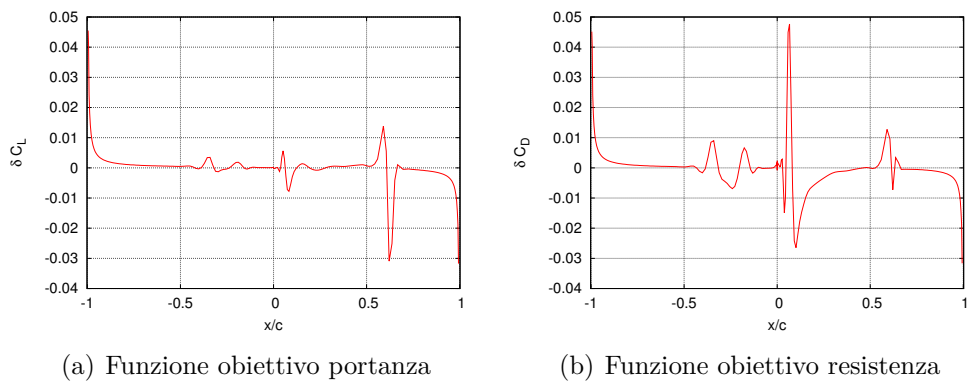


Figura 4.7: Ingrandimento della sensitività sul profilo.

Anche in questo caso il calcolo aggiunto è stato effettuato sulla stessa macchina, il tempo di esecuzione risulta più elevato rispetto al caso diretto ed è di 44.83s per il caso con funzione obiettivo resistenza, mentre per il caso con funzione obiettivo portanza il solutore impiega 49.2s. In Figura 4.8 si riporta l'andamento del residuo dell'equazione di continuità aggiunta.

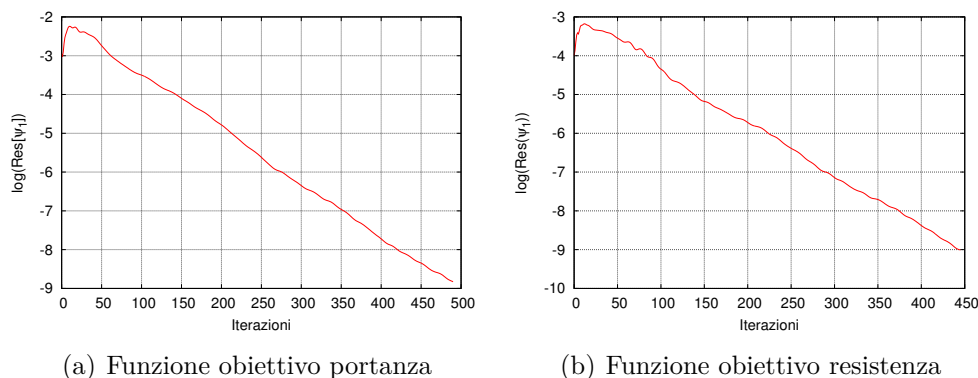


Figura 4.8: Andamento del residuo dell'equazione di continuità aggiunta.

Ottimizzazione geometrica

Avendo a disposizione le sensitività aerodinamiche si può procedere con l'ottimizzazione geometrica. Di seguito sono riportate le richieste dell'ottimizzazione:

- funzione obiettivo: resistenza;
- vincolo di non riduzione della portanza ($C_L \geq 0.3269$).

Le variabili di progetto sono 34 funzioni di Hicks-Henne equispaziate posizionate dal 5% al 95% della corda del profilo su dorso e ventre. La scelta di queste funzioni è la medesima per tutti i casi bidimensionali, in questo modo la deformazione della geometria non può influire sull'angolo d'attacco, che risulta fisso in tutto il processo di ottimizzazione. L'effetto della variazione di angolo d'attacco può essere comunque ottenuto, indirettamente, attraverso la variazione della curvatura della linea media del profilo. Per quanto riguarda la deformazione della griglia di calcolo si è scelto di utilizzare il metodo delle molle lineari. Inoltre bisogna scegliere un criterio di convergenza che in questo caso è basato sul coefficiente di resistenza (per la simulazione diretta) e sull'integrale della sensitività sul profilo (per la simulazione aggiunta); la soluzione si considera a convergenza quando una sequenza di 100 iterazioni del solutore non porta ad una variazione della sesta cifra decimale delle

quantità integrali sopra descritte.

Di seguito si riporta la storia del coefficiente di portanza e resistenza nel processo di ottimizzazione:

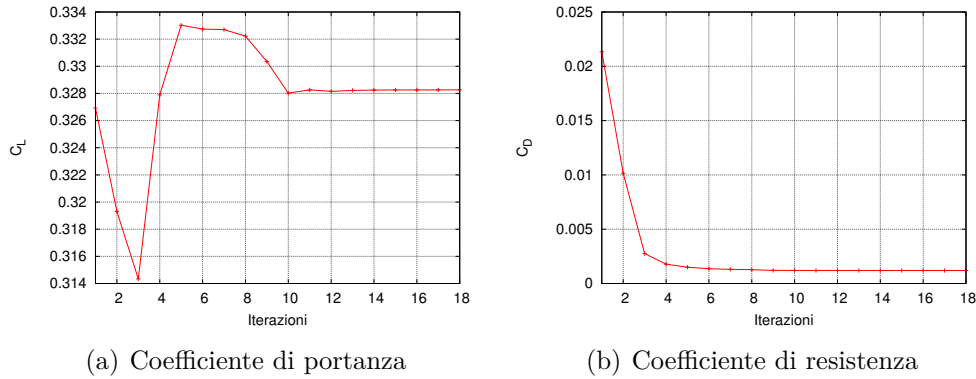


Figura 4.9: Andamento dei coefficienti aerodinamici durante il ciclo di ottimizzazione.

Dopo 18 iterazioni i coefficienti aerodinamici sono i seguenti:

- $C_L = 0.3283$;
- $C_D = 0.0012$.

Il coefficiente di resistenza è diminuito del 94%, mentre il coefficiente di portanza è rimasto sostanzialmente costante.

In Figura 4.10 vengono proposti il numero Mach e il campo di pressione intorno al profilo ottimizzato; il confronto tra l'andamento del coefficiente di pressione del profilo ottimizzato rispetto a quello dell'originale è rappresentato in Figura 4.11.

Dalla Figura 4.11 si può notare che non è presente il tipico plateau dei profili supercritici, ma un comportamento abbastanza simile, infatti il flusso sul dorso è accelerato entro il 10% della corda, dopodichè decelera lentamente; in questo modo si evita la formazione dell'onda d'urto, ottenendo una diminuzione elevata del coefficiente di resistenza. Di seguito, in Figura 4.12, si riporta anche il confronto tra la forma del profilo originale e deformato, si può notare come la superficie sul dorso sia notevolmente appiattita; in questo modo il flusso sul dorso non accelera e si evita la formazione dell'onda d'urto, che è la causa maggiore della resistenza generata dal profilo.

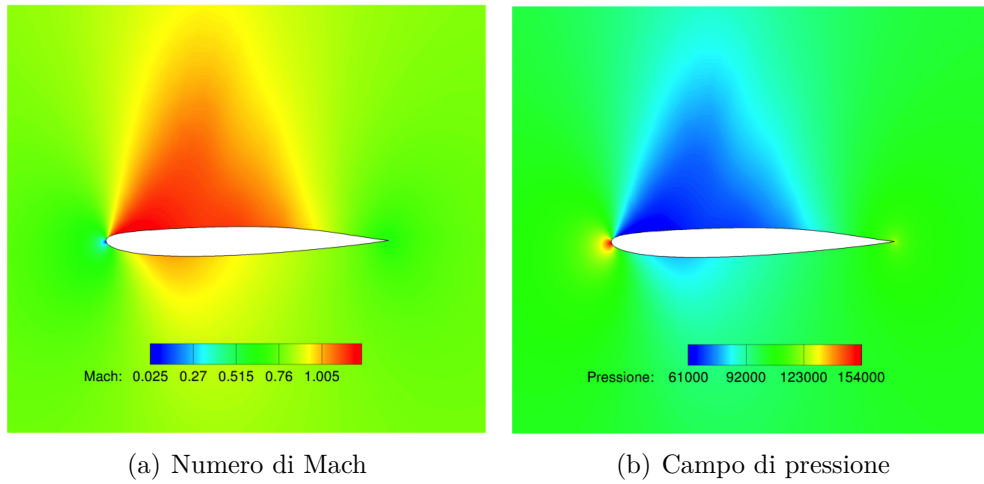


Figura 4.10: Soluzione intorno al NACA 0012 ottimizzato.

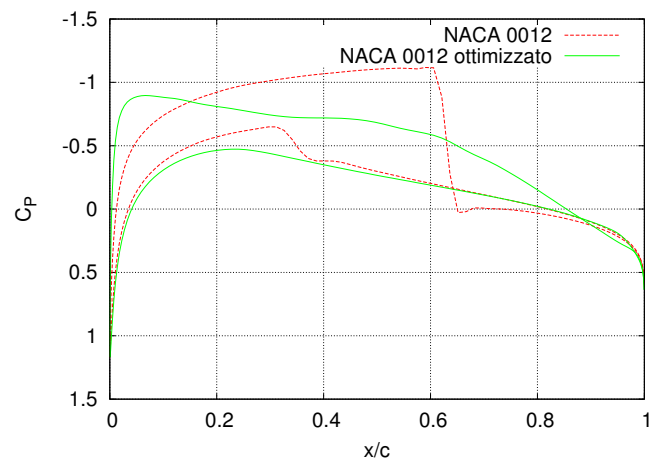


Figura 4.11: Coefficiente di pressione sul profilo originale e ottimizzato.

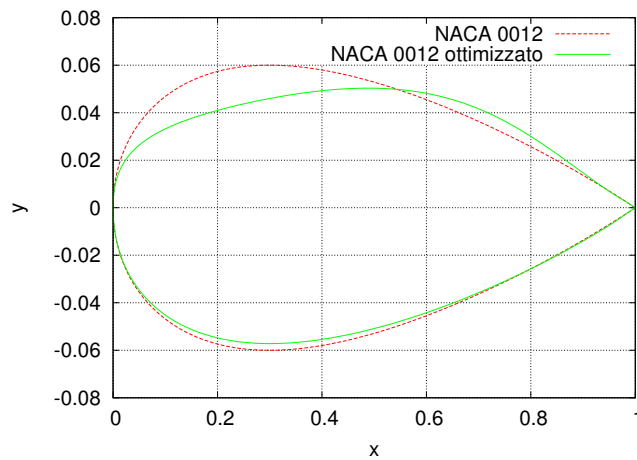


Figura 4.12: Deformazione del profilo.

4.2.2 Caso 2: ONERA M6

Si effettua l'ottimizzazione geometrica di un'ala tridimensionale in condizioni di flusso non viscoso in regime transonico. L'ONERA M6 rappresenta uno standard per le validazioni dei codici di CFD su geometrie tridimensionali; è un'ala a freccia con un basso *aspect ratio* che utilizza il profilo simmetrico ONERA D. Le caratteristiche geometriche dell'ala utilizzata nella simulazione sono:

- apertura alare: $b = 15.14$ m;
- aspect ratio: $\Lambda = 3.8$;
- angolo di freccia: $\Phi = 30^\circ$.

La mesh è non strutturata e formata da 582752 elementi tetraedrici; il dominio è a forma di parallelepipedo ed è rappresentato in Figura 4.13(a). Sull'ala ci sono 36454 elementi triangolari con maggiore raffinatezza all'estremità della stessa, come mostrato in Figura 4.13(b).

Calcolo diretto

Si è scelto di riprodurre il caso test 2308 del riferimento [19] in cui si impongono le seguenti condizioni del flusso all'infinito:

- Mach = 0.835;
- $\alpha = 3.06^\circ$;

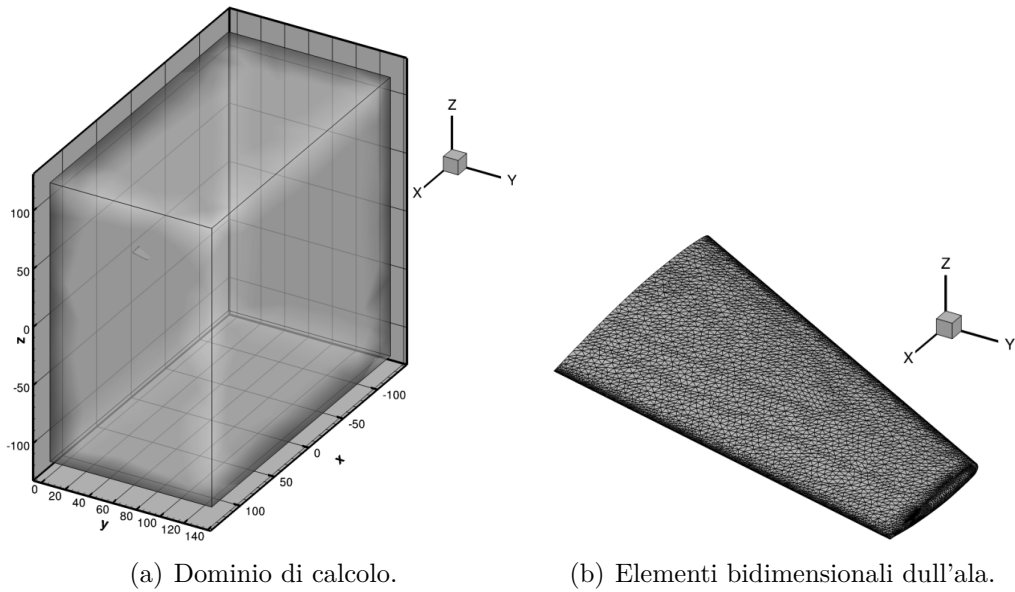


Figura 4.13: Dominio di calcolo ed elementi di griglia sull'ala.

- $P_\infty = 101\,325$ Pa;
- $T_\infty = 273.15$ K.

Nella Figura 4.14 si riportano il coefficiente di pressione e il numero di Mach sull'ala nelle condizioni elencate in precedenza.

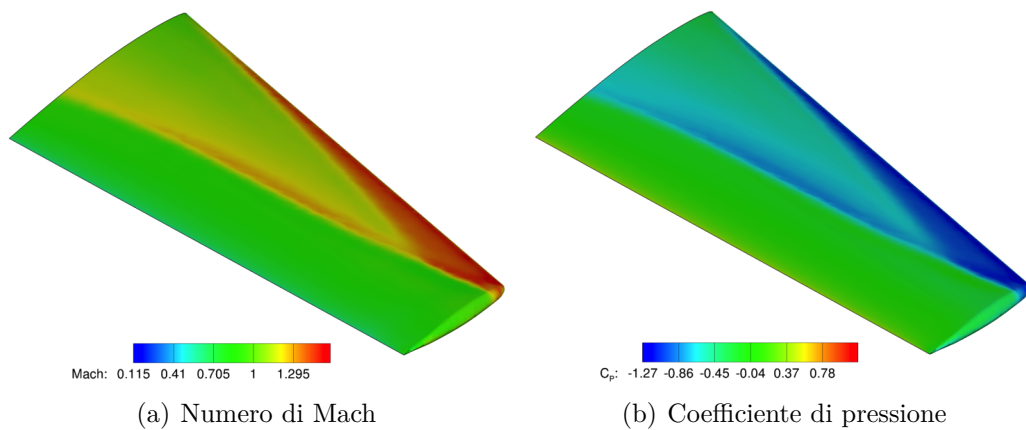


Figura 4.14: Soluzione sull'ala.

Si può notare come siano presenti le tipiche onde d'urto a lambda che generano la parte maggiore di resistenza generata dall'ala; si riporta inoltre il

coefficiente di pressione, a differenti stazioni in apertura dell'ala, confrontato con i dati di riferimento nella Figura 4.15.

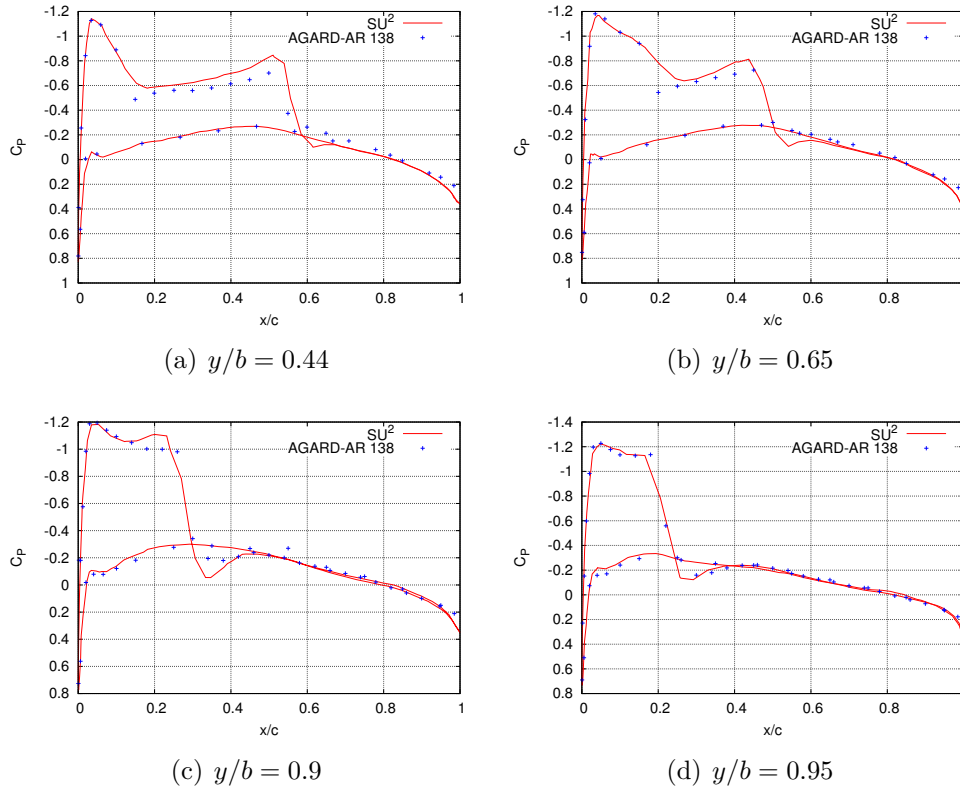


Figura 4.15: Coefficiente di pressione sull'ala ottenuto con SU^2 e confrontato con i dati di riferimento [19].

I coefficienti aerodinamici calcolati sono:

- $C_L = 0.2864$;
- $C_D = 0.0118$.

In questo caso il tempo di calcolo della soluzione è pari a 20 min.

Calcolo aggiunto

Come per il caso precedente si effettuano due simulazioni aggiunte con differenti funzioni obiettivo (resistenza e portanza). Nella figura 4.16 sono riportate le sensitività alle variabili di progetto per le due simulazioni, dove le variabili di progetto saranno definite nel prossimo paragrafo. È possibile notare come, ancora una volta, la sensibilità maggiore sia concentrata in

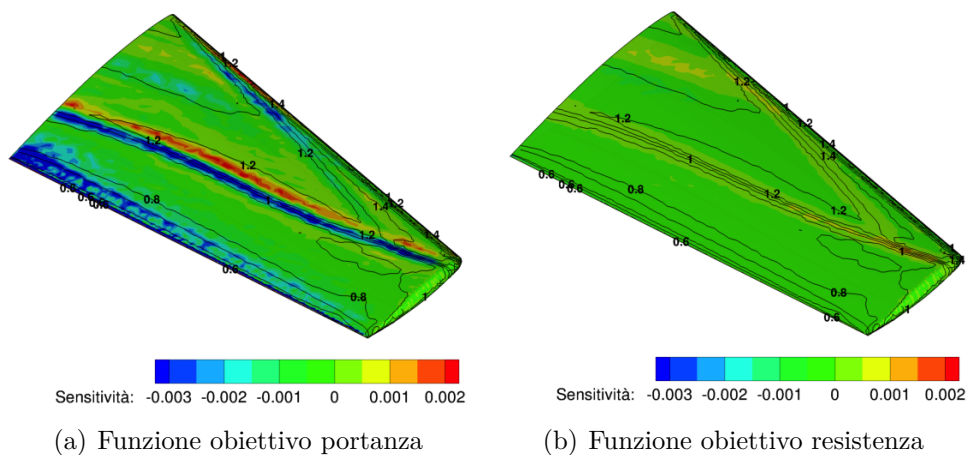


Figura 4.16: Sensitività geometrica alle variabili di progetto e contour del numero di Mach sull'ala.

prossimità del bordo d'uscita e delle onde d'urto. Il tempo di esecuzione delle due simulazioni è pari rispettivamente a 26 min e 29 min.

Ottimizzazione geometrica

Una volta ottenute le sensitività aerodinamiche dalle simulazioni precedenti, si può procedere all'ottimizzazione geometrica.

La prima scelta da fare è rappresentata dalle variabili di design; come già detto, per il caso tridimensionale si usa il metodo FFD. Questo metodo implementato nel software permette, attraverso la definizione di un box esterno, di definire i punti di controllo sul volume da deformare. In questo caso il box è rappresentato in Figura 4.17 e si può notare che non ingloba tutta l'ala, ma solo la zona interessata dalle onde d'urto; la scelta di non proseguire fino al bordo d'uscita è legata al fatto che si vuole verificare che la deformazione tramite il metodo FFD implementato mantenga la connessione e la regolarità tra la parte deformata e indeformata dell'ala, in pratica ci si aspetta che non si perda la connessione tra gli elementi sull'ala.

Il programma necessita solamente della definizione del box esterno (gli 8 punti che definiscono i vertici del prisma) e il grado dei polinomi nelle tre direzioni, poi, richiamando il codice SU2_MDC il software genera in automatico i punti di controllo e le coordinate parametriche della superficie. Nel caso in esame si è scelto di utilizzare un polinomio di primo grado in direzione z , un polinomio di quinto grado in direzione x e di quarto in direzione y .

Come nel caso precedente si richiede :

- funzione obiettivo: resistenza;

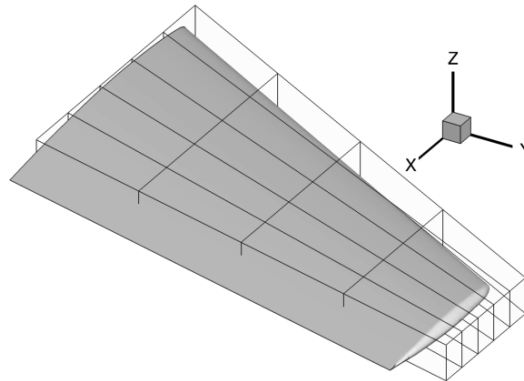


Figura 4.17: Box delle variabili di progetto.

- vincolo di non uguaglianza sulla portanza ($C_L \geq 0.2864$).

Per la deformazione della griglia si è scelto ancora una volta di utilizzare il metodo delle molle, mentre il criterio di convergenza è lo stesso del caso precedente.

Di seguito si riportano l'andamento del coefficiente di portanza e resistenza durante il processo di ottimizzazione:

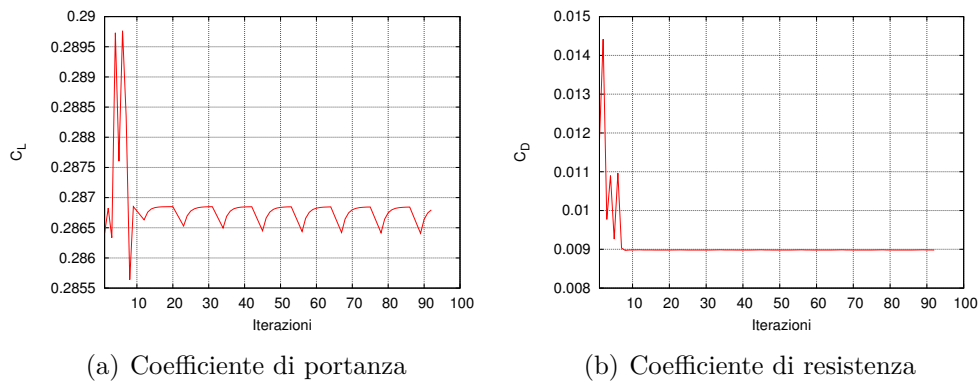


Figura 4.18: Andamento dei coefficienti aerodinamici durante il ciclo di ottimizzazione.

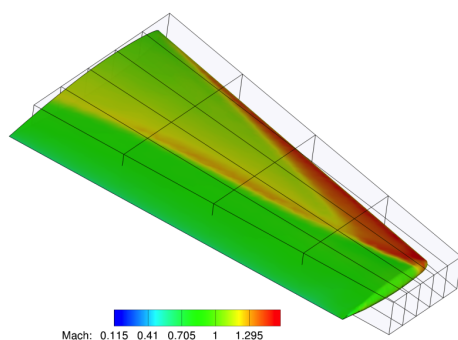
Alla conclusione del ciclo i coefficienti aerodinamici dell'ala ottimizzata sono:

- $C_L = 0.2868$;
- $C_D = 0.00898$.

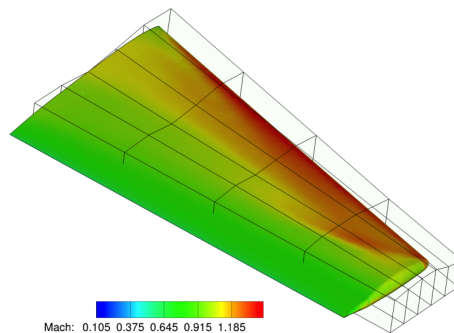
Nel processo si è arrivati ad una diminuzione del coefficiente di resistenza del 24%, mentre il coefficiente di portanza, come nel caso precedente, non è

diminuito.

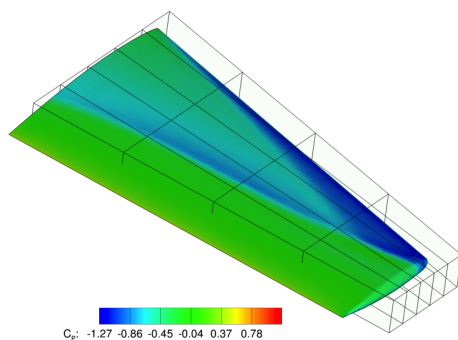
Nella Figura 4.19 sono proposti il campo di pressione e del numero di Mach sull'ala indeformata e ottimizzata, si riporta inoltre la deformazione del box che rende chiaro il meccanismo di funzionamento del metodo FFD.



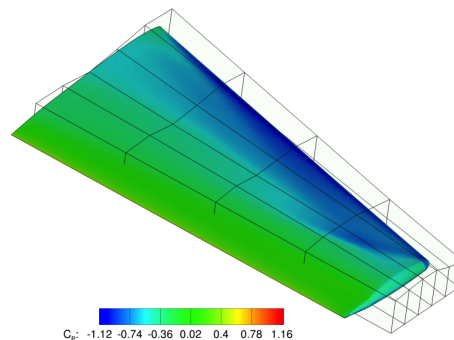
(a) Numero di Mach - ONERA M6



(b) Numero di Mach - ala ottimizzata



(c) Pressione - ONERA M6



(d) Pressione - ala ottimizzata

Figura 4.19: Confronto del numero Mach, del coefficiente di pressione e della deformazione dell'ala.

Inoltre il confronto del coefficiente di pressione con l'ala indeformata, nelle stesse posizioni in apertura precedenti, è proposto in Figura 4.20. Dalle figure si può notare come le onde d'urto non siano più presenti sul dorso dell'ala e la continuità tra la superficie deformata e indeformata è stata mantenuta nel processo di deformazione, infatti la mesh è rimasta regolare in tutte le deformazioni del processo di ottimizzazione.

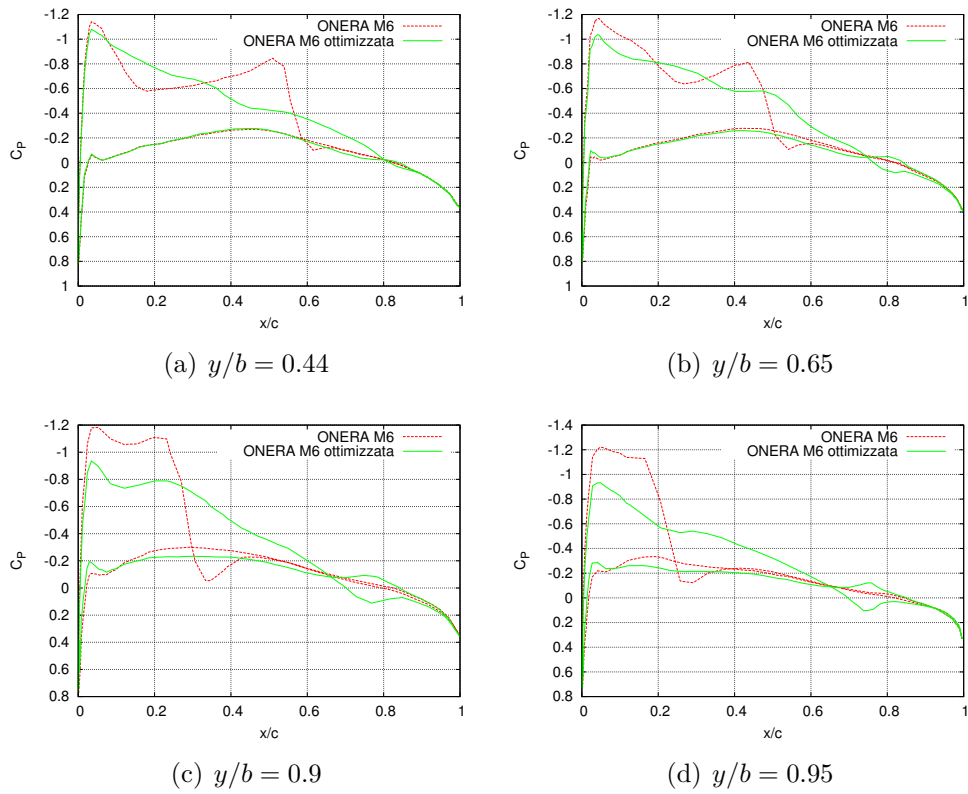


Figura 4.20: Confronto tra il coefficiente di pressione dell'ala indeformata e dell'ala ottimizzata.

4.3 Simulazioni viscosi

In questa sezione vengono proposti solo casi in geometria bidimensionale. Nel primo caso l'ottimizzazione è effettuata per un profilo in regime di moto subsonico e flusso laminare. Nel secondo caso si introduce il modello di turbolenza di Spalart-Allmaras per analizzare un caso in regime transonico e flusso turbolento. Questi due casi viscosi sono molto meno favorevoli degli altri perché il solutore delle equazioni di Navier-Stokes ed il modello di turbolenza sono stati introdotti in SU^2 solo da qualche mese, al contrario del solutore per le equazioni di Eulero, e hanno quindi presentato parecchi problemi di funzionamento.

4.3.1 Caso 3: NACA 0012

La simulazione è stata effettuata su una mesh strutturata a C, formata da 37050 elementi e mostrata in Figura 4.21; il profilo è un NACA 0012 con corda di lunghezza unitaria e gli estremi del dominio sono posti a 100 corde di lunghezza dal profilo. Il profilo è stato discretizzato utilizzando 277 elementi e la cella al bordo d'uscita non ha spessore; nella Figura 4.22 è rappresentata la mesh intorno al profilo e sul bordo d'uscita.

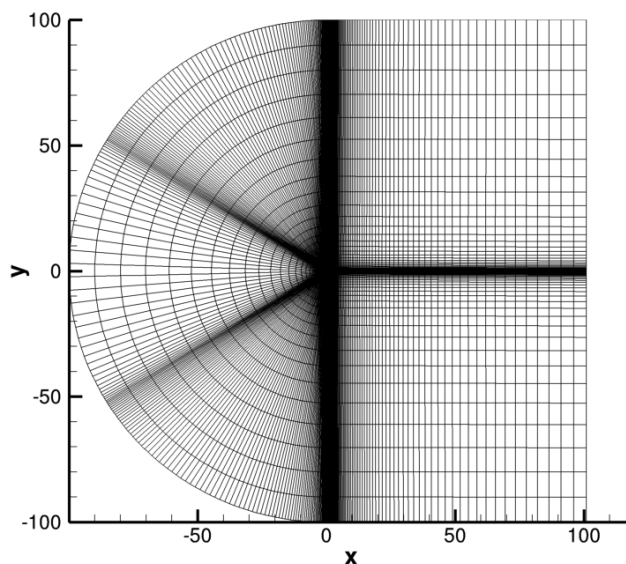


Figura 4.21: Dominio di calcolo.

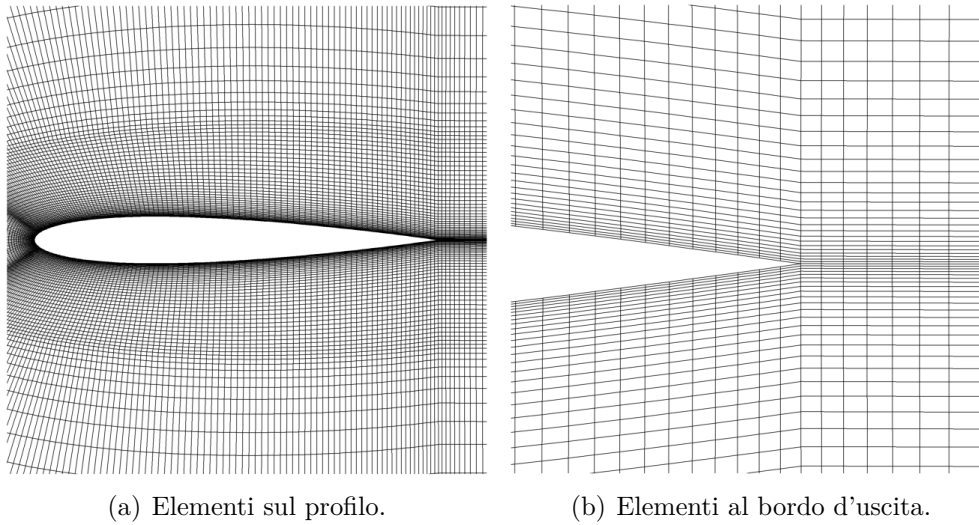


Figura 4.22: Griglia di calcolo intorno al NACA0012.

Calcolo diretto

La simulazione diretta del flusso intorno al profilo ha le seguenti condizioni del flusso all'infinito, che riproducono il caso di riferimento [5]:

- $Mach = 0.3$;
- $Reynolds = 1000$;
- $\alpha = 2.5^\circ$;
- $T = 273.15 \text{ K}$.

In Figura 4.23 si riporta il numero di Mach e il campo di pressione intorno al profilo, mentre in Figura 4.24 l'andamento del coefficiente di pressione sul profilo è confrontato con il caso di riferimento.

I coefficienti di portanza e resistenza calcolati sono:

- $C_L = 0.1296$;
- $C_D = 0.1232$.

In Figura 4.25 si riporta l'andamento del residuo durante il calcolo.

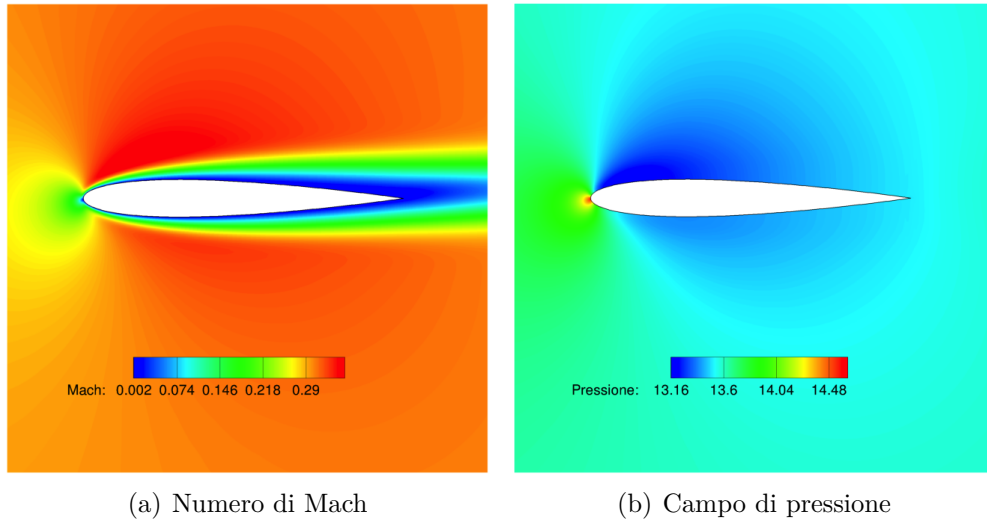


Figura 4.23: Soluzione intorno al NACA 0012.

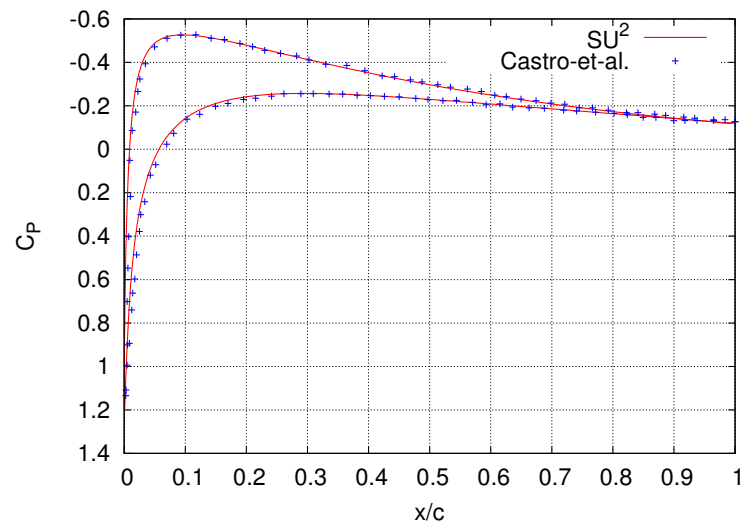


Figura 4.24: Coefficiente di pressione confrontato con il riferimento [5].

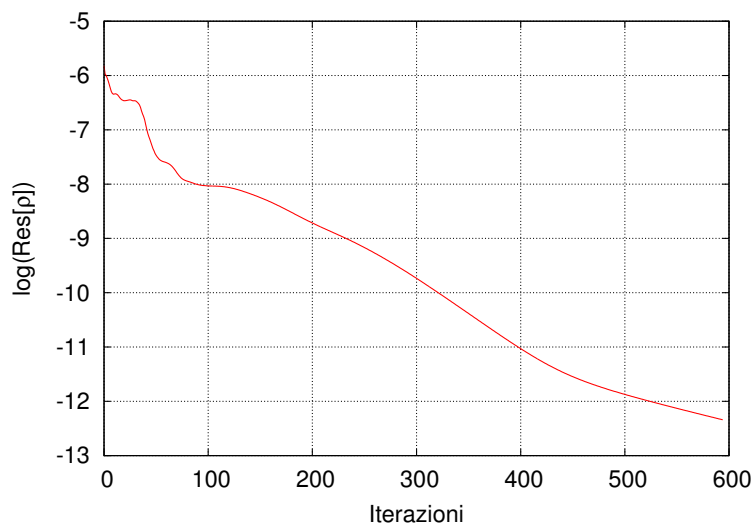


Figura 4.25: Andamento del residuo dell'equazione di continuità.

La macchina utilizzata per il calcolo è la stessa dei casi precedenti e il solutore richiede circa 33 min per il calcolo della soluzione.

Calcolo aggiunto

La soluzione del calcolo diretto ottenuta al passo precedente può essere utilizzata per il calcolo della sensitività aerodinamica tramite il metodo aggiunto. Le condizioni al contorno sono riportate nella Sezione 2.1.1 e non vanno dichiarate dall'utente, ma sono imposte dal solutore una volta note le condizioni al contorno del caso diretto e la funzione obiettivo. Si tenga presente che SU^2 minimizza la funzione obiettivo se essa è rappresentata dalla resistenza, viceversa la massimizza se è rappresentata dalla portanza.

In Figura 4.26 si riporta la componente in direzione x della variabile aggiunta corrispondente alla quantità di moto. Come si vede il campo di moto intorno al profilo è completamente differente dal caso diretto, questo è legato alla particolarità delle condizioni al contorno del metodo aggiunto che prevedono una condizione di *quantità di moto aggiunta* imposta sul profilo. Bisogna tener presente che questa immagine non dice molto sulle zone più sensibili del profilo, difatti la sensitività, come già detto in precedenza, è una funzione molto più complessa delle variabili aggiunte e primarie del flusso.

Il calcolo della sensitività, effettuato come in precedenza sia per la portanza che per la resistenza, conduce ai grafici in Figura 4.27, da leggere con la solita convenzione delle ascisse negative e positive.

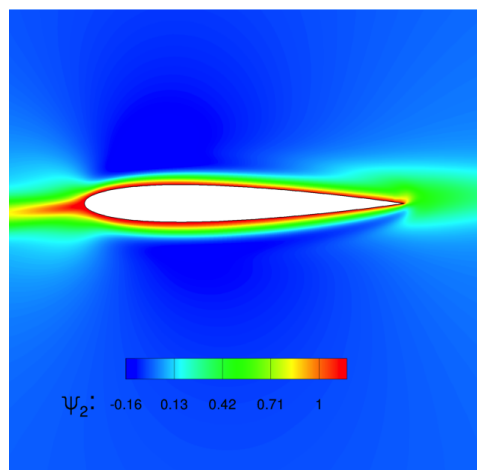


Figura 4.26: Variabile aggiunta Ψ_2 .

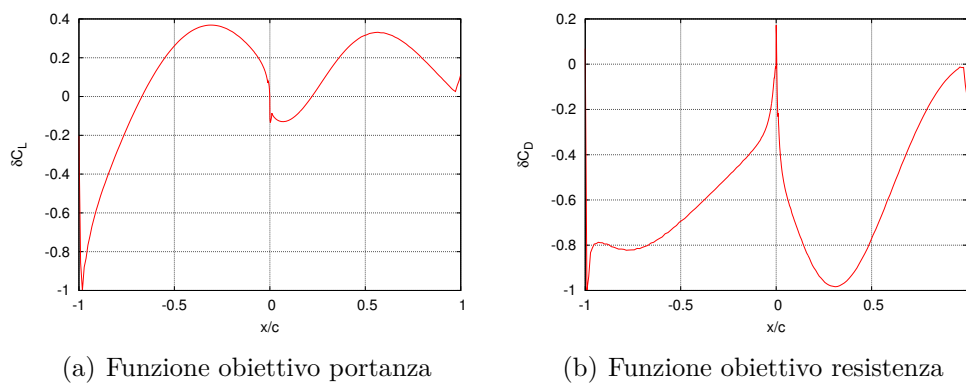


Figura 4.27: Sensitività aerodinamica sul profilo; il semipiano negativo delle ascisse rappresenta il ventre, quello positivo il dorso.

In Figura 4.28 si riportano gli andamenti dei residui della prima equazione aggiunta di entrambe le simulazioni. Si noti come la convergenza sia notevolmente rallentata nel calcolo aggiunto richiedendo molte più iterazioni per raggiungere la convergenza della soluzione. In questo caso il tempo necessario per effettuare 10 iterazioni del solutore è di circa 25 s, comparabile con quello impiegato per la soluzione delle equazioni dirette, ma il numero di iterazioni molto maggiore porta a tempi di calcolo molto elevati.

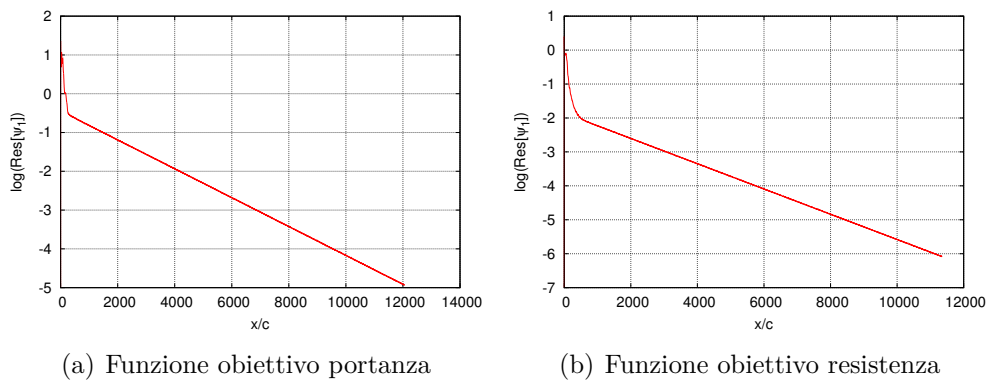


Figura 4.28: Andamento del residuo dell'equazione di continuità aggiunta.

Ottimizzazione geometrica

Si prosegue ora con l'ottimizzazione geometrica con i seguenti requisiti:

- funzione obiettivo: resistenza;
- vincolo di non diminuzione della portanza ($C_L \geq 0.1296$).

Le variabili di progetto sono le stesse del caso inviscido bidimensionale e la deformazione della mesh avviene mediante il metodo delle molle torsionali, che si è rivelato più appropriato nei casi con mesh strutturate. Per quanto riguarda il criterio di convergenza, si è scelto un criterio di convergenza diverso dalle precedenti simulazioni. In questo caso infatti la soluzione si considera a convergenza quando si ha una diminuzione di 6 ordini di grandezza del residuo della prima diretta e aggiunta; questo renderà il ciclo di ottimizzazione molto lento perché, come si è visto dalle simulazioni aggiunte precedenti, la convergenza è notevolmente più lenta rispetto al caso diretto.

I coefficienti aerodinamici al termine del processo di ottimizzazione sono:

- $C_D = 0.119$;
- $C_L = 0.145$.

Questi risultati sono relativi a soli tre passi di ottimizzazione, dopo i quali si sono riscontrati problemi relativi alla deformazione della geometria che ha generato deformazioni troppo elevate e quindi superfici non più riconducibili ad un profilo. Questo problema può essere superato scalando maggiormente la funzione obiettivo e il vincolo, che si traduce in una deformazione minore della geometria. Per quanto il processo iterativo non possa dirsi concluso, il valore del coefficiente aerodinamico di portanza rispetta il vincolo e quello di resistenza è diminuito.

La deformazione finale è rappresentata in Figura 4.29, mentre in Figura 4.30 è riportato il confronto tra il coefficiente di pressione del profilo originale e quello modificato.

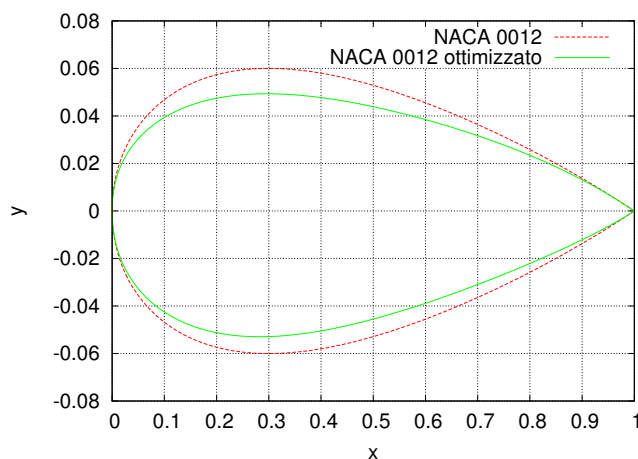


Figura 4.29: Deformazione del profilo.

La deformazione di geometria ha generato una diminuzione del coefficiente di resistenza di circa il 3%, mentre si è ottenuto un aumento del coefficiente di portanza del 12%; questi dati sono in controtendenza con quelli del caso inviscido. Infatti in quel caso si è arrivati ad una diminuzione molto maggiore del coefficiente di resistenza, mentre il coefficiente di portanza è aumentato di pochi punti percentuali. Questa differenza può sicuramente essere legata al fatto che i due casi sono non solo differenti dal punto di vista del fenomeno fisico, ma anche dal punto di vista matematico, nel senso delle equazioni in gioco. Questa affermazione può essere più chiara se si divide il coefficiente di resistenza nel contributo viscoso e in quello di pressione.

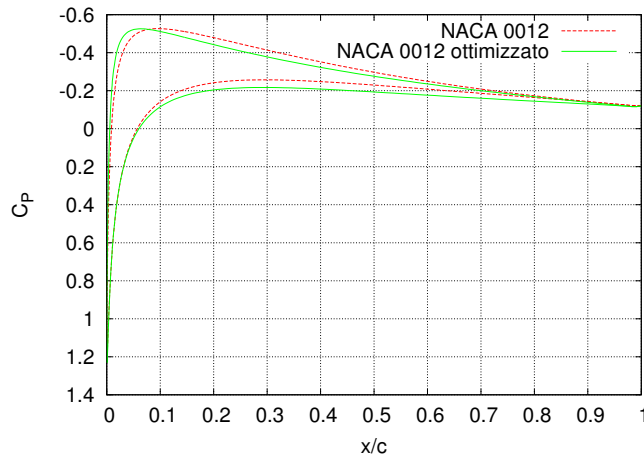


Figura 4.30: Coefficiente di pressione del profilo originale e deformato.

	C_{D_f}	C_{D_p}	C_D
NACA 0012	0.0891	0.0341	0.1232
NACA 0012 ottimizzato	0.0891	0.0299	0.119
	0%	-12%	-3.4%

Tabella 4.1: Scomposizione del coefficiente di resistenza nei due contributi dovuti all'attrito (C_{D_f}) e alla pressione (C_{D_p}).

In Tabella 4.3.1 si può vedere come il contributo maggiore della resistenza è dato dalla componente d'attrito e che questa componente rimanga sostanzialmente invariata, mentre la riduzione della resistenza è tutta spostata sulla componente di pressione; alla luce di ciò si può quindi affermare che la riduzione della resistenza dovuta alla pressione è del 12%.

4.3.2 Caso 4: NACA 0012

È stata utilizzata ancora una volta una mesh a C, ma il dominio è stato ristretto, ponendo i limiti della griglia di calcolo a 30 corde dal profilo e ottenendo la mesh di 56760 elementi mostrata in Figura 4.31. Questa riduzione delle dimensioni del dominio è stata necessaria per evitare che il numero delle celle crescesse enormemente. Infatti in SU^2 non sono (ancora) implementate le *wall functions*, quindi per l'uso di un modello di turbolenza è necessario avere la prima cella di parete al di sotto della prima unità di parete e un rateo di espansione delle celle in direzione ortogonale alla parete non troppo eleva-

to; in questo modo vi è un passaggio graduale delle celle attraverso le diverse regioni dello strato limite turbolento. A seguito della risoluzione maggiore in prossimità della parete, è stato necessario aumentare il numero di elementi sul profilo, in questo caso 527, per evitare che gli elementi al bordo d'uscita avessero un allungamento troppo elevato.

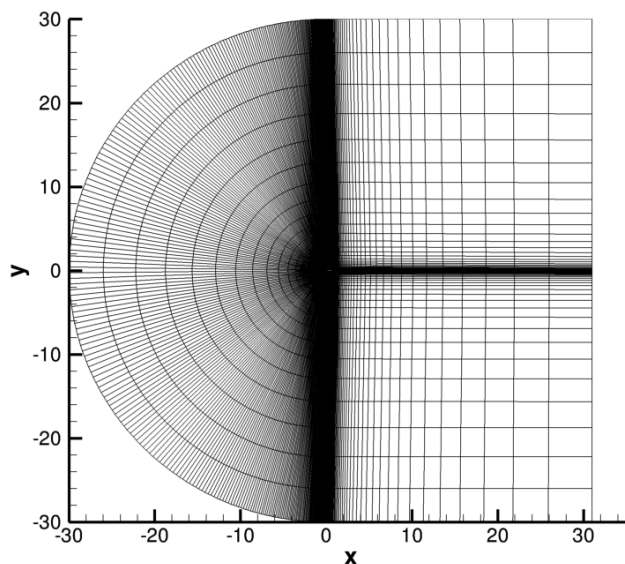


Figura 4.31: Dominio di calcolo.

In Figura 4.32 sono riportati gli elementi intorno al profilo e un ingrandimento al bordo d'uscita, dove la cella del bordo d'uscita non ha spessore.

Calcolo diretto

Il calcolo diretto è stato effettuato con le seguenti condizioni del flusso all'infinito che riproducono il caso di riferimento [19]:

- $\text{Mach} = 0.753$;
- $\text{Reynolds} = 3.88 \cdot 10^6$;
- $\alpha = 1.95^\circ$;
- $T = 273.15 \text{ K}$.

In Figura 4.33 si riporta il numero di Mach e il campo di pressione intorno al profilo, mentre in Figura 4.34 il coefficiente di pressione confrontato con il caso di riferimento.

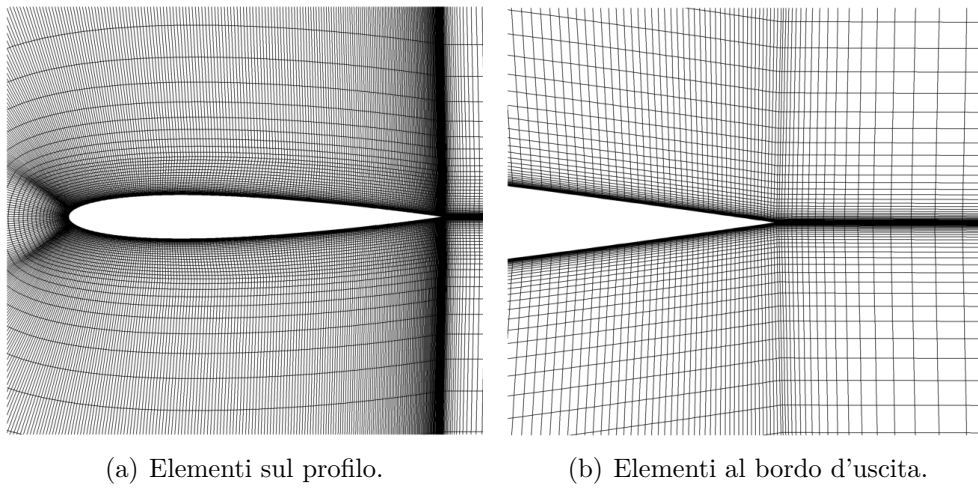


Figura 4.32: Griglia di calcolo intorno al NACA0012.

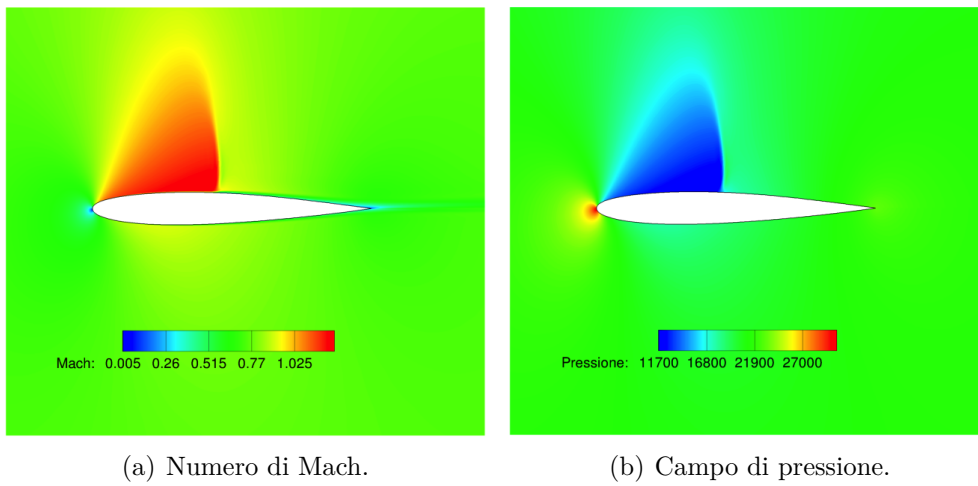


Figura 4.33: Soluzione intorno al profilo.

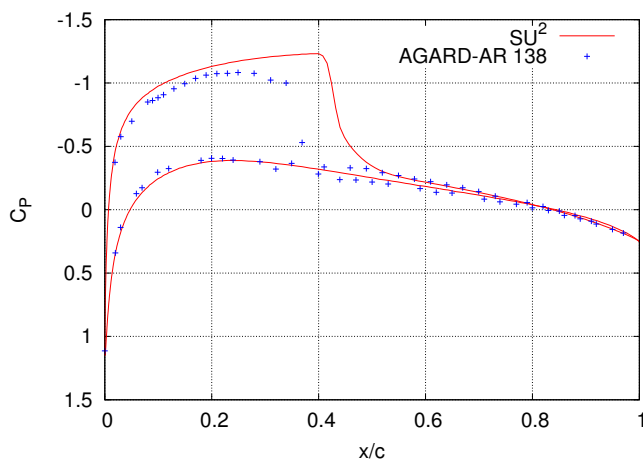


Figura 4.34: Coefficiente di pressione confrontato con il caso di riferimento [19].

I valori dei coefficienti aerodinamici ottenuti sono:

- $C_L = 0.3608$;
- $C_D = 0.01973$.

Dalla Figura 4.34 si può osservare come il coefficiente di pressione non segua fedelmente il riferimento, infatti, nella simulazione effettuata, l'onda d'urto non solo è posizionata più a valle sul dorso, ma ha anche intensità maggiore. Al di là del confronto però, si è interessati ad osservare le zone più sensibili sul profilo e verificare che la procedura di ottimizzazione funzioni; in particolare ci si aspetta che, una delle zone più sensibili, sia in prossimità dell'onda d'urto.

I tempi di calcolo della simulazione diretta sono di poco maggiori di 1 h.

Calcolo aggiunto

Ancora una volta, ottenuta la soluzione delle equazioni dirette, si può procedere alla simulazione aggiunta necessaria per il calcolo della sensibilità alle variabili di progetto, che saranno le stesse dei casi bidimensionali precedenti. In Figura 4.35 si riporta la sensibilità sul profilo, con funzione obiettivo resistenza e portanza. Dall'andamento della sensibilità si osserva come le zone più sensibili del profilo siano in prossimità dell'onda d'urto, proprio come ci si aspettava; altra zona di elevata sensibilità è la zona del dorso in prossimità del bordo d'attacco. A differenza del caso inviscido, possiamo osservare che non è più il bordo d'uscita a presentare la sensibilità maggiore.

Il tempo necessario per la soluzione aggiunta è di circa 3 h 30 min

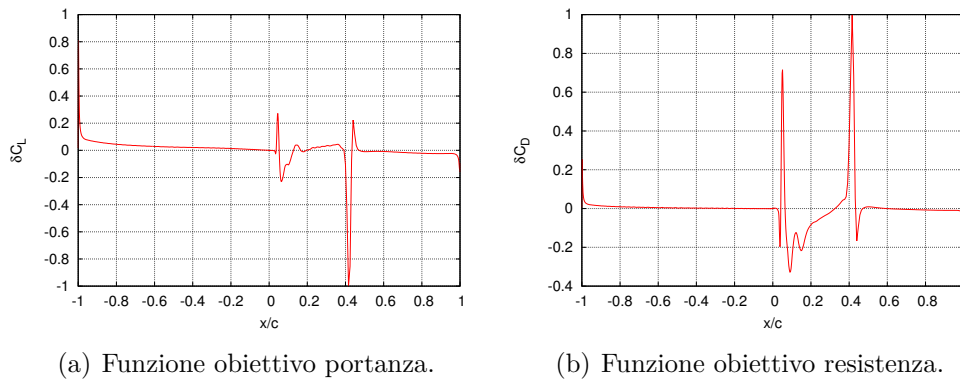


Figura 4.35: Sensitività aerodinamica sul profilo; il semipiano negativo delle ascisse rappresenta il ventre, quello positivo il dorso.

Ottimizzazione geometrica

Come per i casi precedenti, per l'ottimizzazione vengono fatte le seguenti richieste:

- funzione obiettivo: resistenza;
- vincolo di non riduzione della portanza ($C_L \geq 0.3608$).

Le variabili di progetto sono le stesse degli altri casi bidimensionali, mentre per la deformazione della griglia è stato utilizzato il metodo delle molle torsionali. In questa simulazione si torna ad utilizzare come metodo di convergenza quello visto nei casi non viscosi, basato sulla convergenza delle quantità integrali sul profilo.

In questa simulazione, la griglia di calcolo di dimensioni maggiori e l'introduzione del modello di turbolenza rallentano notevolmente il codice, richiedendo parecchio tempo per concludere anche solo un ciclo di ottimizzazione. L'ottimizzatore, che si è fermato dopo 5 iterazioni del ciclo, a causa ancora una volta di una deformazione troppo elevata, ha portato ai seguenti risultati:

- $C_L = 0.3676$;
- $C_D = 0.01513$.

Mentre in Figura 4.36 è riportata l'andamento dei coefficienti in funzione dei passi dell'ottimizzatore. Si può notare come si sia arrivati ad una diminuzione del coefficiente di resistenza del 23% e un aumento del coefficiente di portanza di circa il 2%. Anche in questo caso, come nella simulazione viscosa precedente, è utile suddividere il contributo della resistenza in componente

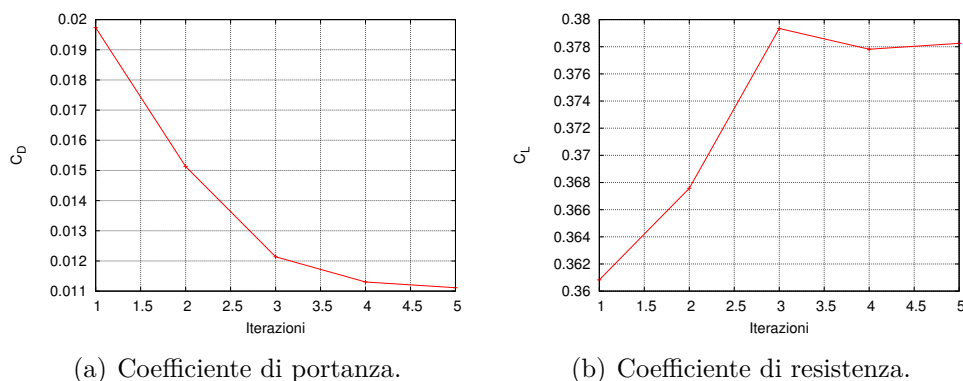


Figura 4.36: Andamento dei coefficienti aerodinamici nel ciclo di ottimizzazione.

	C_{D_f}	C_{D_P}	C_D
NACA 0012	0.0065	0.0132	0.0197
NACA 0012 ottimizzato	0.0068	0.0083	0.0151
	+4.6%	-37%	-23%

Tabella 4.2: Scomposizione del coefficiente di resistenza nei due contributi dovuti all'attrito (C_{D_f}) e alla pressione (C_{D_P}).

dovuta alla pressione e componente dovuta all'attrito viscoso; a tale proposito i dati sono riportati in Tabella 4.3.2.

Si può osservare come ora, in regime turbolento, la componente che contribuisce di più alla resistenza sia la componente di pressione; si mette in evidenza inoltre che la componente dovuta all'attrito sia rimasta praticamente invariata dopo la deformazione. Ancora una volta, facendo riferimento alla sola componente di pressione si ha una forte riduzione della resistenza. Infatti in Figura 4.37 si può notare come sia stata ridotta l'intensità dell'onda d'urto, generando così, una riduzione di resistenza rilevante nonostante il ciclo di ottimizzazione non sia a convergenza. Inoltre si riportano, in Figura 4.33, il numero di Mach e il campo di pressione intorno al profilo e in Figura 4.39 la deformazione del profilo.

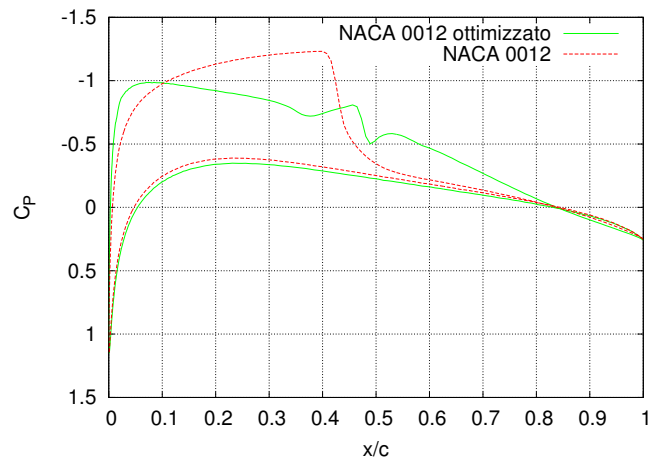
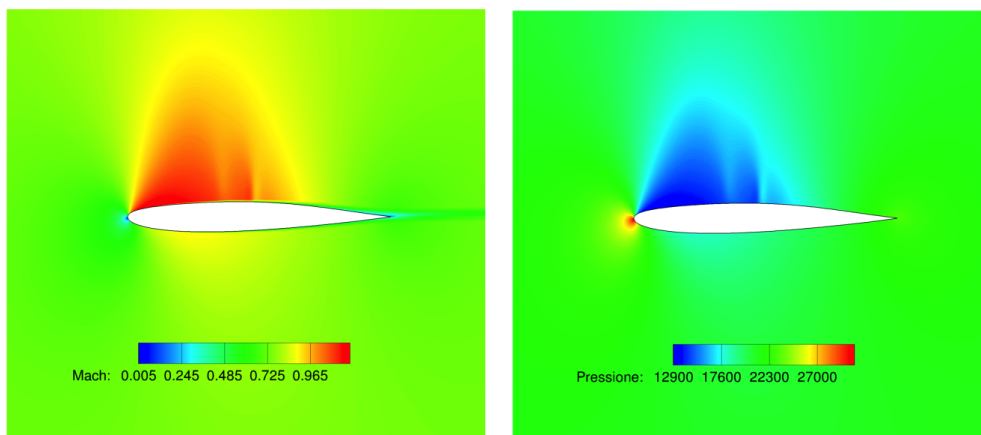


Figura 4.37: Coefficiente di pressione sul profilo indeformato e profilo ottimizzato.



(a) Numero di Mach.

(b) Campo di pressione.

Figura 4.38: Soluzione intorno al profilo ottimizzato.

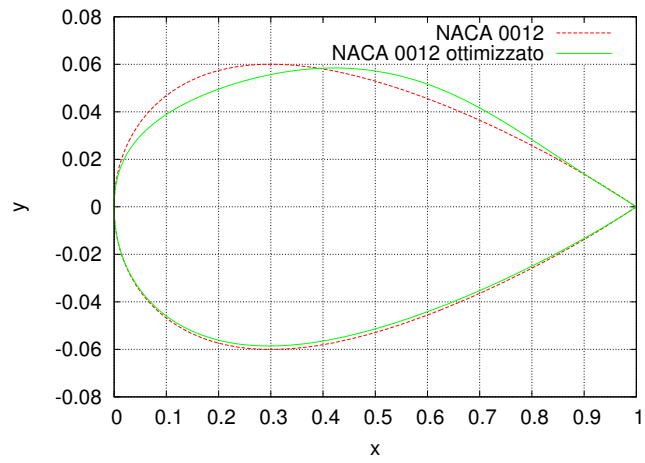


Figura 4.39: Deformazione del profilo.

Conclusioni

Questo lavoro ha per la prima volta messo alla prova il codice CFD SU^2 ed in particolare la sua capacità di offrire soluzioni al problema dell'ottimizzazione di forma facendo ricorso alla soluzione delle equazioni aggiunte.

Sono stati analizzati casi differenti dal punto di vista della modellazione matematica e dal punto di vista della fisica in gioco. Dalle analisi effettuate si può affermare che il metodo aggiunto continuo è senza dubbio un valido strumento per evidenziare le zone più sensibili alla variazione di geometria di un corpo investito da un fluido; inoltre tutta l'implementazione dello strumento che effettua l'*ottimizzazione geometrica* funziona e rende il processo interamente automatizzato.

Nei casi analizzati si abbiamo sempre ottenuto una riduzione del coefficiente di resistenza; nei casi viscosi tuttavia la riduzione si è rivelata essere tutta relativa alla componente di pressione, mentre la componente viscosa è sempre rimasta pressoché invariata.

Per quanto riguarda la deformazione della superficie nel processo di ottimizzazione si può affermare che sia nel caso bidimensionale che tridimensionale non si sono riscontrati grossi problemi, se non nella deformazione dei Casi 3 e 4 in cui le deformazioni andrebbero scalate ulteriormente per evitare deformazioni elevate. Nel caso tridimensionale, la scelta di implementare il metodo FFD è stata favorevole perchè rende molto più semplice la definizione, da parte dell'utente, della superficie, o parte della superficie, da ottimizzare.

In conclusione si può affermare che il software presenta buone qualità e ottime potenzialità, anche se ha una storia ancora troppo recente e diversi aspetti da migliorare, tali da renderlo per ora non adatto all'utilizzo in ambito industriale. D'altra parte, i problemi e le difficoltà incontrate sono stati innumerevoli. L'efficienza computazionale, in particolar modo per il solutore aggiunto, è piuttosto bassa. La possibilità di funzionare in parallelo, aggiunta recentemente, al momento non funziona. Il solutore delle equazioni di Navier-Stokes, specialmente in combinazione con il modello di turbolenza, ha un funzionamento problematico.

Bibliografia

- [1] Implementing turbulence models into the compressible rans equations. <http://turbmodels.larc.nasa.gov/implementrans.html>.
- [2] W. Kyle Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28:443–480, 1999.
- [3] J. Blazek. *Computational fluid dynamics: principles and applications*. 2001.
- [4] Alfonso Bueno-Orovio, Carlos Castro, Francisco Palacios, and Enrique Zuazua. Continuous adjoint approach for the spalart-allmaras model in aerodynamic optimization. *AIAA Journal*, 50(3):631–646, 2012.
- [5] Carlos Castro, Lozano Carlos, Francisco Palacios, and Zuazua Enrique. A systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. *AIAA Journal*, 45(9):2125–2139, 2007.
- [6] Christoph Degand and Charbel Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computer and Structures*, 80:305–316, 2002.
- [7] C. Farhat, C. Degand, B Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163:231–245, 1998.
- [8] Giovanni Fasano. *Cenni sui metodi di ottimizzazione vincolata*, 2008.
- [9] A. Favre. Equations des gaz turbulents compressibles. *Journal de Mecanique*, 4(3):361–390, 1965.
- [10] Michael B. Giles and Niles A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.

- [11] J. Hadamard. *Leçons sur le Calcul des Variations*. Gauthier-Villars, Paris, 1910.
- [12] D. Kraft. Software package for sequential quadratic programming. Technical report, Hannover, 1988.
- [13] C.L. Lawson and B.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [14] Francisco Palacios, Michael R. Colonno, Aniket C. Aranake, Alejandro Campos, Sean R. Copeland, Thomas Economon, Amrita K. Lonkar, Trent W. Lukaczyk, Thomas W. R. Taylor, and Juan J. Alonso. Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. *AIAA*, 2013.
- [15] O. Pironneau. An introduction to the adjoint approach to design. *Journal of Fluid Mechanics*, 64:97–110, 1964.
- [16] Stephen B. Pope. *Turbulent Flows*. 2011.
- [17] R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial Value Problems*. Wiley-Interscience, London, 1967.
- [18] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *ACM*, 20(4):151–160, 1986.
- [19] Autori vari. *Experimental Data Base for Computer Program Assessment*. AGARD Advisory Group of Aerospace Research and Development NATO, 1979. AGARD-AR-138.
- [20] Autori vari. *Test Cases for Inviscid Flow Field Methods*. AGARD Advisory Group of Aerospace Research and Development NATO, 1985. AGARD-AR-211.
- [21] D.C. Wilcox. *Turbulence Modeling for CFD*. 1993.
- [22] R.B. Wilson. *A Simplicial Algorithm for Concave Programming*. PhD thesis, Harvard University, 1963.