

UNIVERSITÀ degli STUDI di GENOVA



**SCUOLA POLITECNICA
DIPARTIMENTO DI INGEGNERIA MECCANICA,
ENERGETICA, GESTIONALE
E DEI TRASPORTI (DIME)**

**Corso di Laurea Magistrale in Ingegneria Meccanica - Energia e
Aeronautica**

**Identificazione di profili aerodinamici
da dati di scia: un approccio di Inverse Design
tramite il metodo dell'aggiunto discreto**

Relatori:

Prof. Ing. Pralits Jan Oscar
Prof. Ing. Guerrero Rivas Joel

Laureando:

Anelli Filippo
Matricola: 4943875

Co-Relatori:

Ing. Segalerba Eric
Ing. Di Fabbio Tony

Sommario

Questa tesi affronta il problema dell'*inverse design* aerodinamico: la ricostruzione della forma di un corpo a partire dalla misura del campo di velocità nella sua scia, mediante ottimizzazione basata sul metodo dell'aggiunto discreto. Il tema nasce dall'interesse, sia scientifico sia ingegneristico, di decodificare la firma di scia, accessibile a tecniche di misura non intrusive come la *Particle Image Velocimetry*, per risalire alla geometria che l'ha generata, in un problema notoriamente mal posto e sensibile alla configurazione di osservazione. L'attività si è articolata in due fasi. Nella prima, il metodo è stato sviluppato e caratterizzato in dettaglio su un corpo tozzo bidimensionale (deformazione di un'ellisse verso un cilindro, in regime laminare), con uno studio parametrico sistematico della posizione, del numero e della densità dei probes, della funzione obiettivo e del numero di Reynolds. Nella seconda, è stato esteso alla ricostruzione di un profilo alare NACA in regime turbolento. I risultati mostrano che la qualità della ricostruzione è governata dal contenuto informativo della scia più che dalla quantità di dati, che il valore dell'obiettivo è disaccoppiato dall'accuratezza geometrica, e che esistono limiti fisici di osservabilità legati alla posizione dei sensori e alla geometria di partenza.

Abstract

This thesis addresses the problem of aerodynamic *inverse design*: the reconstruction of a body's shape from measurements of the velocity field in its wake, through optimization based on the discrete adjoint method. The topic is motivated by the scientific and engineering interest in decoding the wake signature, accessible to non-intrusive measurement techniques such as *Particle Image Velocimetry*, in order to recover the geometry that generated it, in a problem that is notoriously ill-posed and sensitive to the observation setup. The work was carried out in two stages. In the first, the method was developed and characterized in detail on a two-dimensional bluff body (deformation of an ellipse into a cylinder, in the laminar regime), through a systematic parametric study of probe position, number and density, of the objective function, and of the Reynolds number. In the second, it was extended to the reconstruction of a NACA airfoil in the turbulent regime. The results show that reconstruction quality is governed by the information content of the wake rather than by the amount of data, that the objective value is decoupled from geometric accuracy, and that physical observability limits exist, tied to sensor placement and initial geometry.

Indice

1	Introduzione	1
1.1	Il problema inverso in fluidodinamica	1
1.2	Stato dell'arte	2
1.2.1	Inverse design classico	2
1.2.2	Assimilazione di dati	2
1.2.3	Machine learning e reti neurali	3
1.3	Ottimizzazione di forma basata sul gradiente e metodo aggiunto	3
1.3.1	Origini del metodo aggiunto in fluidodinamica	3
1.4	Strumenti open-source per l'ottimizzazione di forma	4
1.5	Obiettivi e struttura della tesi	4
2	Impostazione del problema	6
2.1	Equazioni di governo	6
2.2	Funzione obiettivo	7
2.2.1	Configurazione dei probes	7
2.2.2	Definizione della funzione obiettivo	7
2.2.3	Formulazione variazionale e vincolo NS	8
3	Metodologia di ottimizzazione	9
3.1	Derivazione dell'aggiunto discreto	9
3.2	Soluzione dell'equazione aggiunta: GMRES e approccio matrix-free	10
3.2.1	Il metodo GMRES	10
3.2.2	Approccio matrix-free	11
3.2.3	Precondizionamento	11
3.3	Calcolo delle derivate parziali	11
4	Metodologia Computazionale e Strumenti	13
4.1	Architettura del framework e flusso di calcolo	13
4.1.1	Il risolutore DASimpleFoam	14
4.1.2	Gli ottimizzatori: SLSQP e IPOPT	14
4.2	Free Form Deformation	15
4.2.1	Il box FFD per il cilindro	15
4.2.2	Propagazione della deformazione alla mesh	16
4.3	Setup dell'ambiente di calcolo	16
4.3.1	Containerizzazione con Docker	16
4.3.2	Parallelizzazione MPI	17
4.3.3	Script di automazione	17
5	Impostazione del Caso di Studio	18
5.1	Struttura generale del caso DAFoam e automazione tramite <code>runScript.py</code>	18
5.2	Generazione e deformazione della mesh: configurazione	20
5.2.1	Geometria iniziale e target	20
5.2.2	Topologia e parametri della mesh	20

5.2.3	Condizioni al contorno	23
5.2.4	Schema di soluzione e parametri numerici SIMPLE	23
5.2.5	Deformazione della mesh con IDWarp	23
5.3	Implementazione della Funzione Obiettivo	24
5.3.1	Preparazione del campo di riferimento	24
5.3.2	Approccio preliminare: modulo della differenza vettoriale	24
5.3.3	Implementazione della funzione VarianceScaled	24
5.3.4	Configurazione generale dei probes e varianti	26
5.3.5	Ottimizzatore IPOPT: configurazione	27
6	Analisi Parametrica e Studio di Fattibilità	29
6.1	Indipendenza dalla griglia computazionale	29
6.2	Metrica di valutazione della ricostruzione geometrica	31
6.2.1	Scelta della metrica e significato dei tre indicatori	32
6.2.2	Origine dell'errore residuo: i limiti della parametrizzazione FFD	32
6.3	Studio parametrico del posizionamento dei probes	33
6.3.1	Panoramica dei sei casi	33
6.3.2	Disaccoppiamento tra obiettivo e accuratezza geometrica	34
6.3.3	Effetto del numero di probes	37
6.3.4	Effetto della posizione: scia intermedia contro scia lontana	42
6.4	Instabilità nella scia prossima	43
6.4.1	Meccanismo dell'instabilità: sensibilità strutturale, inversione di segno e amplificazione	45
6.5	Analisi di sensibilità al numero di Reynolds	48
6.5.1	Interpretazione fisica della dipendenza dal Reynolds	49
6.5.2	Effetto della densità dei probes	50
6.6	Confronto della funzione obiettivo: Variance contro VarianceScaled	51
6.6.1	Meccanismo della superiorità di VarianceScaled	54
6.7	Casi estremi: probe molto lontane e colonna singola	55
6.7.1	Identificabilità: quanti sensori servono davvero	58
6.7.2	Il ruolo dell' <i>inverse crime</i> : portata e limiti dei risultati estremi	59
6.8	Sintesi dell'analisi parametrica	60
6.8.1	Linee guida di progetto	61
7	Risultati dell'Inverse Design sul Corpo Tozzo	63
7.1	Convergenza e ricostruzione geometrica del caso di riferimento	63
7.2	Confronto qualitativo dei profili ricostruiti	65
7.3	Il campo aggiunto e la struttura della sensibilità	66
7.4	Struttura del campo fluidodinamico e della scia	68
7.4.1	Convergenza del campo di errore di velocità	70
7.5	Residui, convergenza dell'aggiunto e costi computazionali	72
7.6	Sintesi	73

8	Applicazione Avanzata – Inverse Design di un Profilo Alare	74
8.1	Introduzione	74
8.2	Dati di riferimento e configurazione target	75
8.2.1	Costruzione del campo di riferimento (UData)	75
8.3	Geometrie di partenza	76
8.3.1	Generazione della mesh	76
8.3.2	Nota sul bordo di fuga tronco	76
8.4	Metodo numerico	77
8.4.1	Equazioni di governo e solutore	77
8.4.2	Analisi di sensitività aggiunta	77
8.4.3	Funzione obiettivo	77
8.4.4	Parametrizzazione della geometria (FFD)	77
8.4.5	Vincoli geometrici	78
8.4.6	Algoritmo di ottimizzazione	78
8.5	Configurazione del flusso e della chiusura turbolenta	79
8.5.1	Scelta del numero di Reynolds e compromesso sulla mesh di parete	79
8.6	Finestra di osservazione e strategia di posizionamento dei probes	80
8.6.1	Finestra di base e traslazione a valle	80
8.6.2	Generazione delle coordinate dei probes	81
8.6.3	Metrica di errore geometrico e soglia di arresto	82
8.7	Risultati	82
8.7.1	Panoramica	82
8.7.2	Effetto della posizione streamwise della finestra	82
8.7.3	NACA0015/W1: quasi-ottimo transitorio e divergenza geometrica	88
8.7.4	Effetto della geometria di partenza	92
8.7.5	Comportamento di convergenza	92
8.8	Discussione	93
8.9	Conclusioni	93
8.9.1	Posizionamento raccomandato dei probes e geometria migliore	94
9	Conclusioni e Sviluppi Futuri	95
9.1	Risultati sul corpo tozzo	95
9.2	Risultati sul profilo alare	96
9.3	Considerazioni metodologiche trasversali	97
9.4	Limitazioni	97
9.5	Sviluppi futuri	97
A	Script di automazione dell'ottimizzazione: runScript.py	99
B	Calcolo dell'errore geometrico: geometric_error.py	107
C	Post-processing della convergenza: plottingtot.py	113
D	Campo di velocità e condizioni al contorno: file 0/U	116
	Bibliografia	117

Capitolo 1

Introduzione

La capacità di inferire la geometria di un corpo a partire dalla misura di grandezze fluidodinamiche nel suo campo di scia è un problema di grande rilevanza sia scientifica che ingegneristica. In aerodinamica, la *firma di scia*, cioè il profilo di velocità, pressione o vorticità che persiste a valle di un corpo immerso in un fluido, contiene informazioni codificate sulla forma che l'ha generata. La possibilità di decodificare queste informazioni apre la strada a tecniche di *inverse design* capaci di ricostruire la geometria di un corpo senza accesso diretto alla sua superficie, basandosi unicamente su misure nel campo di moto a valle. Tale problema appartiene alla vasta classe dei *problemi inversi* in meccanica dei fluidi, classe caratterizzata in generale dall'essere mal posta nel senso di Hadamard: la soluzione può non essere unica, la sua esistenza non è garantita e può essere estremamente sensibile ai dati in ingresso [1].

Le motivazioni pratiche sono molteplici. In aerodinamica sperimentale, tecniche di misura ottiche non intrusive come la *Particle Image Velocimetry* (PIV) e la *Laser Doppler Velocimetry* (LDV) consentono di ricostruire il campo di velocità in ampie regioni del dominio fluido senza perturbarne il comportamento, ma non forniscono direttamente la geometria del corpo. Se da un insieme di misure di scia fosse possibile ricostruire la forma, si aprirebbe la possibilità di caratterizzare oggetti sommersi, di identificare deformazioni strutturali in volo, o di eseguire controllo di qualità su componenti aerodinamici senza richiederne la misurazione diretta. In campo ambientale, analoghe considerazioni si applicano alla stima della forma di ostacoli naturali (colline, edifici) a partire da misurazioni anemometriche a valle, con applicazioni alla modellazione del vento urbano e alla dispersione di inquinanti. In campo biomeccanico, la ricostruzione della forma di cetacei o uccelli da misurazioni nel loro scia ha stimolato ricerche sul nuoto e sul volo ad alta efficienza.

1.1 Il problema inverso in fluidodinamica

Il problema diretto della fluidodinamica computazionale (CFD) consiste nel calcolare il campo di moto attorno a una geometria data. Il problema inverso rovescia questa relazione: nota una misura (parziale e rumorosa) del campo di moto, si vuole determinare la geometria, o più in generale un insieme di parametri del sistema, che è compatibile con tale misura. La formulazione matematica generale è la seguente: dato un operatore non lineare $\mathcal{F}$ che mappa la geometria Ω nello spazio delle osservabili $\mathbf{d}$,

$$\mathbf{d} = \mathcal{F}(\Omega) + \varepsilon, \quad (1.1)$$

dove ε è il rumore di misura, si cerca Ω tale che $\mathcal{F}(\Omega)$ si avvicini il più possibile alle osservazioni disponibili. Nella presente tesi, $\mathcal{F}$ è il risolutore RANS stazionario che calcola il campo di velocità $\mathbf{U}$ attorno al corpo, e le osservazioni $\mathbf{d}$ sono i valori di velocità campionati da un insieme di sensori (*probes*) distribuiti nella scia.

La mal-posizione del problema si manifesta in modo acuto nel caso della ricostruzione da scia: geometrie anche assai diverse possono produrre firme di scia molto simili, specialmente se le misure sono concentrate in una piccola regione del dominio. Come

mostrato recentemente da Zhang e Karniadakis [1], la consistenza del modello di turbolenza adottato è determinante per la qualità della ricostruzione: inconsistenze tra il modello usato per generare i dati di riferimento e quello usato nel ciclo di ottimizzazione inversa possono tradursi in differenze fino al 250% nelle sensitività geometriche, rendendo la convergenza impossibile o fuorviante. Un ulteriore fattore di regolarizzazione consiste nell'impiegare misurazioni a più condizioni operative (ad angoli di attacco diversi), il che riduce drasticamente il numero di soluzioni compatibili [1].

1.2 Stato dell'arte

La letteratura sul problema inverso in aerodinamica è ampia e si sviluppa lungo tre principali filoni metodologici: l'*inverse design* classico, l'*assimilazione di dati*, e i metodi basati sull'*apprendimento automatico*.

1.2.1 Inverse design classico

Il filone storicamente più consolidato è l'*inverse design* classico, in cui si cerca la geometria che produce una distribuzione di pressione o di circolazione prescritta sulla superficie del corpo. I primi lavori risalgono all'aerodinamica potenziale: il metodo di Lighthill (1945) e le tecniche basate sulla *conformal mapping* (es. trasformazione di Joukowski) permettono, in regime incomprimibile e irrotazionale, di costruire profili con una distribuzione di pressione target. Il *panel method* inverso implementato nel codice XFOIL di Drela [2] estende queste idee al regime viscoso-strato limite in modo interattivo, ed è ancora oggi lo strumento di riferimento per la progettazione rapida di profili alari a basso numero di Reynolds. In regime transonico, l'*inverse design* diretto diventa molto più complesso a causa della presenza di onde d'urto: tecniche basate sulla *residual-correction* e sul *fictitious gas* sono state sviluppate per affrontare questo regime, ma richiedono in genere la disponibilità del campo di pressione superficiale, non del campo di velocità a monte. Un limite comune a questi approcci classici è che l'informazione che guida il processo di inversione si trova sulla superficie del corpo, mentre nel caso di interesse per questa tesi le osservazioni sono nella scia, a distanza dal corpo. Questo complica notevolmente il problema, perché il legame tra la geometria e la firma di scia è mediato dall'intera soluzione del campo di moto: nel caso laminare, dalle equazioni di Navier–Stokes stazionarie; nel caso turbolento, dalle equazioni mediate alla Reynolds (RANS) con un opportuno modello di chiusura.

1.2.2 Assimilazione di dati

L'assimilazione di dati (*data assimilation*, DA) è un insieme di tecniche nate in meteorologia e oceanografia per fondere osservazioni sperimentali con modelli fisici al fine di stimare sia lo stato del sistema sia parametri non misurabili direttamente [3]. I due approcci principali sono il metodo variazionale 4D-Var e il filtro di Kalman d'insieme (EnKF).

Il metodo 4D-Var risolve un problema di controllo ottimale: si minimizza una funzione costo che misura la distanza tra le previsioni del modello e le osservazioni disponibili su una finestra temporale, soggetto al vincolo che la soluzione soddisfi le equazioni del modello. Il gradiente della funzione costo viene calcolato efficientemente tramite l'equazione aggiunta del modello [4]. In ambito fluidodinamico, Artana et al. [5] hanno applicato uno schema di assimilazione variazionale alla ricostruzione di flussi viscosi

instazionari da dati PIV, dimostrando la capacità di recuperare il campo completo di velocità a partire da misure sparse. Lavori più recenti hanno esteso questi approcci alla stima di parametri del modello di turbolenza [6] e alla ricostruzione di flussi turbolenti in parete [7].

Il filtro di Kalman d'insieme (EnKF) approssima la distribuzione di probabilità dello stato del sistema mediante un insieme (*ensemble*) di realizzazioni del modello. È particolarmente adatto per sistemi non lineari in cui il calcolo dell'aggiunto è difficoltoso, e ha trovato applicazione nella stima di parametri di modelli RANS [8]. Il limite principale dell'EnKF per applicazioni CFD ad alta fedeltà è l'elevato costo computazionale legato alla necessità di eseguire un gran numero di simulazioni dell'insieme.

1.2.3 Machine learning e reti neurali

Negli ultimi anni, i metodi basati sull'apprendimento automatico hanno trovato crescente applicazione nella ricostruzione e nell'ottimizzazione di flussi aerodinamici. Due approcci principali si distinguono: i modelli *data-driven* puri e le *Physics-Informed Neural Networks* (PINNs).

I modelli *data-driven* addestrano reti neurali profonde (convoluzionali, ricorrenti, o basate su *attention*) su grandi database di coppie (geometria, campo di moto) generate tramite simulazioni CFD, al fine di apprendere la mappa forma-flusso in modo implicito [9]. Una volta addestrate, queste reti possono produrre previsioni a costo computazionale trascurabile, rendendole attraenti per applicazioni real-time. Tuttavia, la generalizzazione al di fuori del dominio di addestramento rimane un problema aperto, e la qualità delle previsioni degrada rapidamente in regime di flusso non visto durante il training.

Le *Physics-Informed Neural Networks* (PINNs) [10] rappresentano un approccio ibrido: la rete neurale deve soddisfare non solo i dati osservati, ma anche le equazioni differenziali alle derivate parziali che governano il sistema, incluse come termine di penalità nella funzione di perdita. Le PINNs sono state applicate con successo a problemi inversi in fluidodinamica [11], inclusa l'identificazione di parametri di flusso e di confini geometrici [12]. Il loro principale vantaggio è la capacità di lavorare su dati sparsi e rumorosi senza richiedere una mesh strutturata. Il loro principale limite è la difficoltà di scalare a numeri di Reynolds elevati e a geometrie complesse, dove le equazioni di Navier–Stokes sviluppano strutture spaziali molto fini che la rete neurale fatica a rappresentare.

1.3 Ottimizzazione di forma basata sul gradiente e metodo aggiunto

Parallelamente ai filoni descritti sopra, un approccio distinto e di grande maturità tecnologica è l'ottimizzazione di forma basata sul gradiente, in cui le sensitività della funzione obiettivo rispetto ai parametri geometrici sono calcolate in modo rigoroso e usate per guidare iterativamente la modifica della geometria verso un ottimo. L'efficienza di questo approccio dipende in modo critico dalla possibilità di calcolare il gradiente a basso costo, indipendentemente dal numero di variabili di progetto.

1.3.1 Origini del metodo aggiunto in fluidodinamica

L'applicazione del metodo aggiunto all'ottimizzazione di sistemi governati da equazioni differenziali risale al lavoro pionieristico di Lions [13] nel quadro del controllo ottimale di equazioni alle derivate parziali. In ambito fluidodinamico, Pironneau [14] derivò per

primo le equazioni aggiunte delle equazioni di Stokes e le applicò al problema di minima resistenza, mostrando che il corpo di minima resistenza a parità di volume ha un bordo d'attacco a spigolo di 90° .

L'estensione al caso comprimibile e al regime transonico fu realizzata da Jameson [15], che nel 1988 formulò le equazioni aggiunte per le equazioni di Eulero e le applicò al progetto aerodinamico inverso e diretto di profili alari transonici. Questo lavoro segnò una svolta nella progettazione aerodinamica computazionale, rendendo per la prima volta possibile l'ottimizzazione di forma ad alta fedeltà con un numero di simulazioni CFD indipendente dalla dimensione dello spazio di progetto. L'estensione alle equazioni di Navier–Stokes con termini viscosi fu successivamente realizzata da Jameson et al. [16] e da Nielsen e Anderson [17], aprendo la strada all'ottimizzazione di forma in regime viscoso e turbolento.

La presente tesi adotta la variante dell'*aggiunto discreto*, in cui l'aggiunto viene derivato direttamente a partire dal sistema di equazioni discretizzato del risolutore CFD, garantendo derivate con precisione al livello della macchina. La derivazione formale e il confronto con l'approccio aggiunto continuo sono trattati nel dettaglio nel Capitolo 3. Per la parametrizzazione della geometria si utilizza la tecnica della *Free Form Deformation* (FFD) [18], che verrà descritta nel Capitolo 4.

1.4 Strumenti open-source per l'ottimizzazione di forma

Il panorama degli strumenti computazionali aperti per l'ottimizzazione aerodinamica si è notevolmente arricchito nell'ultimo decennio. L'intero framework computazionale impiegato in questa tesi è open-source: OpenFOAM [19] funge da risolutore CFD, DAFoam [20] vi si innesta come livello aggiunto discreto e interfaccia con gli ottimizzatori gradient-based, e il framework MACH-Aero coordina parametrizzazione geometrica, deformazione della mesh e ottimizzazione. Come alternativa indipendente si cita SU2 [21], che integra nativamente le capacità adjoint ed è ampiamente adottato in ambito aerospaziale. L'architettura dettagliata di questi strumenti e le scelte implementative adottate sono discusse nel Capitolo 4.

1.5 Obiettivi e struttura della tesi

La presente tesi si inserisce nel contesto dell'inverse design aerodinamico basato sull'aggiunto, con un focus specifico sulla ricostruzione della forma di corpi tozzi e di profili alari a partire da misure di velocità nella scia. L'approccio adottato è quello dell'*inverse crime* consapevole: il campo di riferimento è generato con lo stesso risolutore e la stessa configurazione fisica utilizzati nel ciclo di ottimizzazione inversa, in modo da eliminare qualsiasi incongruenza di modello e isolare l'effetto delle sole variabili di interesse: posizionamento dei probes e geometria iniziale. Nel caso principale, flusso laminare incomprimibile a $Re = 40$ attorno a un corpo tozzo, ciò significa che sia la soluzione di riferimento sia tutte le iterazioni di ottimizzazione risolvono le stesse equazioni di Navier–Stokes laminari. Nel caso dell'applicazione al profilo alare (Capitolo 8), a Reynolds più elevato e regime turbolento, l'approccio viene esteso impiegando lo stesso modello di chiusura Spalart–Allmaras sia per la generazione del campo target sia per l'ottimizzazione. Gli obiettivi principali della tesi sono:

- Implementare e validare una funzione obiettivo personalizzata basata su misure di velocità da probes posizionati nella scia del corpo, in grado di guidare in modo robusto il processo di ottimizzazione inversa.
- Condurre un'analisi parametrica sistematica del numero, della posizione e della distanza dei probes dal corpo, al fine di identificare le configurazioni di misura che garantiscono la migliore ricostruzione geometrica.
- Dimostrare la convergenza dell'algoritmo su una progressione di casi di crescente complessità geometrica: da un corpo tozzo bidimensionale (ellisse verso cilindro) a un profilo alare NACA.
- Analizzare criticamente le limitazioni del metodo, con particolare attenzione agli effetti del posizionamento dei probes rispetto alla bolla di ricircolazione di scia e alla stabilità numerica dell'ottimizzazione.

Il resto del documento è organizzato come segue. Il Capitolo 2 descrive l'impostazione matematica del problema: le equazioni di governo del flusso e la definizione della funzione obiettivo. Il Capitolo 3 illustra nel dettaglio il metodo dell'aggiunto discreto utilizzato per il calcolo delle sensitività. Il Capitolo 4 descrive l'architettura del framework computazionale (OpenFOAM, DAFoam, FFD). Il Capitolo 5 presenta la configurazione del caso di studio e l'implementazione pratica del ciclo di ottimizzazione. Il Capitolo 6 riporta l'analisi parametrica dei probes. Il Capitolo 7 presenta i risultati dell'ottimizzazione sul corpo tozzo e il Capitolo 8 quelli sul profilo alare. Infine, le conclusioni e le prospettive di sviluppo futuro sono raccolte nel Capitolo 9.

Capitolo 2

Impostazione del problema

2.1 Equazioni di governo

Si considera il moto bidimensionale di un fluido attorno a un corpo tozzo (*bluff body*) in condizioni di flusso stazionario e incomprimibile. Il numero di Reynolds del caso di studio principale è $Re = \rho U_\infty D / \mu = 40$, dove D è la lunghezza caratteristica del corpo (diametro del cilindro o asse maggiore dell'ellisse) e U_∞ la velocità di flusso libero. A questo valore di Re il flusso è stazionario e laminare: non si sviluppano instabilità di scia e non è necessario alcun modello di chiusura turbolenta.

Il campo di moto è governato dalle equazioni di Navier–Stokes incomprimibili stazionarie. L'equazione di continuità esprime la conservazione della massa:

$$\nabla \cdot \mathbf{U} = 0, \quad (2.1)$$

mentre l'equazione della quantità di moto esprime il bilancio tra forze di pressione, inerzia e viscosità:

$$\nabla \cdot (\mathbf{U}\mathbf{U}) = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{U}, \quad (2.2)$$

dove $\mathbf{U} = [u, v]^T$ è il vettore velocità bidimensionale, p la pressione cinematica (p/ρ), e $\nu = \mu/\rho$ la viscosità cinematica del fluido. La (2.2) è la forma semplificata valida per un fluido newtoniano incomprimibile a viscosità costante, in cui il tensore degli sforzi viscosi si riduce a $\boldsymbol{\tau} = 2\mu \mathbf{S}$, con $\mathbf{S} = \frac{1}{2}(\nabla \mathbf{U} + (\nabla \mathbf{U})^T)$ il tensore velocità di deformazione.

Il sistema (2.1)–(2.2) è chiuso: le incognite sono il campo di velocità $\mathbf{U}$ e la pressione p , entrambi da determinare sul dominio fluido $\mathcal{D}$ soggetto alle seguenti condizioni al contorno:

- *Ingresso* (Γ_{in}): velocità uniforme $\mathbf{U} = U_\infty \hat{\mathbf{e}}_x$;
- *Uscita* (Γ_{out}): condizione di pressione nulla e gradiente di velocità nullo in direzione normale ($\partial \mathbf{U} / \partial n = \mathbf{0}$, $p = 0$);
- *Pareti laterali* (Γ_{sym}): simmetria o slip, a seconda della configurazione del dominio;
- *Superficie del corpo* ($\partial \Omega$): condizione di no-slip, $\mathbf{U} = \mathbf{0}$.

La soluzione stazionaria di questo sistema fornisce il campo di moto $(\mathbf{U}, p)$ che costituisce il punto di partenza per il calcolo della funzione obiettivo e per la soluzione del problema aggiunto.

Nota sull'applicazione al profilo alare. Nel Capitolo 8, dove si estende la metodologia a un profilo alare NACA a numero di Reynolds significativamente più elevato, il flusso non è più laminare e le equazioni di governo vengono estese al regime turbolento tramite il modello di Spalart–Allmaras. Il modello di chiusura e le relative equazioni aggiuntive saranno introdotti in quel capitolo.

2.2 Funzione obiettivo

Nella formulazione del problema di inverse design adottata in questa tesi, l'obiettivo è trovare la geometria Ω del corpo che produce un campo di velocità quanto più simile possibile a un campo di riferimento $\mathbf{U}^{\text{ref}}$, noto soltanto in un sottoinsieme discreto di punti del dominio fluido. Questi punti sono detti *probes* e costituiscono il legame tra il modello computazionale e le osservazioni sperimentali (o, nel caso di studio di questa tesi, le misure numeriche generate da una simulazione target).

2.2.1 Configurazione dei probes

Si definisce un insieme di N_p probes posizionati in punti $\mathbf{x}_i \in \mathbb{R}^2$, con $i = 1, \dots, N_p$, collocati nella regione di scia a valle del corpo. In ogni punto $\mathbf{x}_i$ si misurano le due componenti del vettore velocità:

$$\mathbf{U}(\mathbf{x}_i) = [u(\mathbf{x}_i), v(\mathbf{x}_i)]^T, \quad (2.3)$$

e analogamente il corrispondente valore di riferimento $\mathbf{U}^{\text{ref}}(\mathbf{x}_i)$. Il campo di riferimento è ottenuto dalla soluzione RANS stazionaria calcolata attorno alla geometria target, e interpolato nelle posizioni dei probes tramite la funzione di mappatura del campo `mapFields` di OpenFOAM, come descritto nel Capitolo 5.

I probes sono disposti secondo una griglia rettangolare nella scia, con un passo spaziale uniforme in direzione sia longitudinale (x) che trasversale (y). La scelta della posizione e del numero di probes è uno dei fattori chiave del problema di inverse design: probes troppo vicini al corpo rischiano di cadere all'interno della bolla di ricircolazione, dove il campo di moto è altamente sensibile alla geometria ma anche molto irregolare, mentre probes troppo lontani a valle misurano una scia già diffusa in cui le informazioni sulla geometria sono parzialmente perse. L'analisi sistematica di questo compromesso è l'oggetto del Capitolo 6.

2.2.2 Definizione della funzione obiettivo

La funzione obiettivo f misura la discrepanza tra il campo di velocità simulato e quello di riferimento alle posizioni dei probes. L'implementazione adottata è il modulo `DAFunctionVarianceScaled` di `DAFoam`, che calcola un *errore relativo quadratico medio* normalizzato puntualmente dal valore locale di riferimento. Per un campo vettoriale $\mathbf{U}$ con componenti selezionate dall'insieme $K \subseteq \{0, 1\}$ (nel caso bidimensionale, $K = \{0, 1\}$ corrisponde alle componenti u e v), la funzione obiettivo è:

$$f(\mathbf{x}, \mathbf{w}) = \frac{s}{N_{\text{ref}}} \sum_{i=1}^{N_p} \sum_{k \in K} \left(\frac{U_k(\mathbf{x}_i; \mathbf{x}, \mathbf{w}) - U_k^{\text{ref}}(\mathbf{x}_i)}{U_k^{\text{ref}}(\mathbf{x}_i)} \right)^2, \quad (2.4)$$

dove s è un fattore di scala numerico, $N_{\text{ref}} = N_p \cdot |K|$ è il numero totale di valori di riferimento (probes per componenti), e $U_k^{\text{ref}}(\mathbf{x}_i)$ è il valore della k -esima componente di velocità nel campo target al probe i . Il termine $(U_k - U_k^{\text{ref}})/U_k^{\text{ref}}$ è l'errore relativo della k -esima componente al probe i : esso è adimensionale e indipendente dall'intensità assoluta del campo di velocità.

Il nome `VarianceScaled` deriva dal fatto che la somma degli errori quadratici è normalizzata ("scalata") componente per componente dal quadrato del valore di riferimento

locale, anziché da una quantità globale come la varianza statistica del campo. Quando il campo simulato riproduce esattamente il riferimento in tutti i probes, $f = 0$; il valore cresce quanto più la geometria corrente si discosta dalla geometria target.

2.2.3 Formulazione variazionale e vincolo NS

Il problema di ottimizzazione inversa si scrive formalmente come:

$$\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{w}(\mathbf{x})) \quad \text{soggetto a} \quad \mathbf{R}(\mathbf{x}, \mathbf{w}) = \mathbf{0}, \quad (2.5)$$

dove $\mathbf{R}(\mathbf{x}, \mathbf{w}) = \mathbf{0}$ rappresenta il sistema delle equazioni di Navier–Stokes discretizzate (2.1)–(2.2), che funge da vincolo di uguaglianza: per ogni configurazione geometrica $\mathbf{x}$, il campo di moto $\mathbf{w}$ deve essere la soluzione stazionaria. Questa è la struttura di un problema di controllo ottimale vincolato da una PDE (*PDE-constrained optimization*), per il quale il metodo dell’aggiunto è lo strumento naturale per calcolarne efficientemente il gradiente, come dettagliato nel Capitolo 3.

La dipendenza di f dalle variabili di progetto $\mathbf{x}$ è sia esplicita, attraverso la posizione dei probes rispetto alla geometria deformata, sia implicita, attraverso il campo di moto $\mathbf{w}(\mathbf{x})$ che risolve il problema per la geometria corrente.

Capitolo 3

Metodologia di ottimizzazione

3.1 Derivazione dell'aggiunto discreto

I gradienti di forma, ovvero le sensitività della funzione obiettivo rispetto ai punti di controllo della FFD, sono calcolati mediante il metodo dell'aggiunto [15, 22]. Questo approccio è particolarmente efficiente quando il numero di variabili di ingresso (punti di controllo FFD) è elevato rispetto al numero di funzioni obiettivo (nel presente lavoro pari a 1). Per garantire la precisione delle derivate, fondamentale per la rapida convergenza dell'ottimizzazione, si adotta la variante dell'aggiunto discreto [22].

Nell'implementazione discreta del metodo aggiunto [22], si assume che una forma discretizzata delle equazioni di governo (Navier–Stokes e, in generale, equazioni di chiusura della turbolenza) sia disponibile in forma di residuo nel risolutore CFD. Si definiscono il vettore delle variabili di progetto $\mathbf{x} \in \mathbb{R}^{n_x}$ e il vettore delle variabili di stato $\mathbf{w} \in \mathbb{R}^{n_w}$ tali che

$$\mathbf{R}(\mathbf{x}, \mathbf{w}) = \mathbf{0}, \quad (3.1)$$

dove $\mathbf{R} \in \mathbb{R}^{n_w}$ è il vettore residuo del sistema discretizzato. La funzione obiettivo f dipende in generale sia dalle variabili di progetto sia da quelle di stato:

$$f = f(\mathbf{x}, \mathbf{w}). \quad (3.2)$$

Il calcolo di f è computazionalmente meno oneroso rispetto al calcolo del vettore residuo $\mathbf{R}$, che contiene termini non lineari. L'analisi di sensitività consiste nel valutare le derivate totali della funzione obiettivo rispetto alle variabili di progetto; poiché $\mathbf{w} = \mathbf{w}(\mathbf{x})$ (il campo di moto dipende implicitamente dalla geometria attraverso le equazioni di governo), è necessario applicare la regola della catena:

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{x}}. \quad (3.3)$$

Il termine $d\mathbf{w}/d\mathbf{x}$ è computazionalmente proibitivo: richiederebbe la risoluzione di n_x sistemi lineari (uno per ciascuna variabile di progetto). Un approccio alternativo sfrutta la derivata totale del vettore residuo: poiché la soluzione deve soddisfare $\mathbf{R}(\mathbf{x}, \mathbf{w}) = \mathbf{0}$ per qualsiasi variazione delle variabili di progetto, anche la derivata totale di $\mathbf{R}$ rispetto a $\mathbf{x}$ deve essere nulla:

$$\frac{d\mathbf{R}}{d\mathbf{x}} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{x}} = \mathbf{0}. \quad (3.4)$$

Combinando le equazioni (3.3) e (3.4) si ottiene un'espressione per la sensitività che non dipende esplicitamente da $d\mathbf{w}/d\mathbf{x}$:

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} - \underbrace{\frac{\partial f}{\partial \mathbf{w}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{w}} \right)^{-1}}_{\psi^T} \frac{\partial \mathbf{R}}{\partial \mathbf{x}}, \quad (3.5)$$

dove le derivate parziali nell'equazione (3.5) possono essere valutate tramite differenziazione algoritmica [23]. L'inverso di $\partial \mathbf{R} / \partial \mathbf{w}$ non viene calcolato esplicitamente: si

introduce invece il vettore aggiunto ψ , definito dalla relazione identificata nella (3.5), e lo si determina risolvendo il sistema lineare

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right)^T \psi = \left(\frac{\partial f}{\partial \mathbf{w}}\right)^T, \quad (3.6)$$

che prende il nome di *equazione aggiunta*. Con la soluzione di (3.6), la sensitività finale assume la forma compatta:

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} - \psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \quad (3.7)$$

Il vantaggio computazionale è decisivo: il costo del calcolo della sensitività completa $df/d\mathbf{x}$ è indipendente dalla dimensione n_x del vettore delle variabili di progetto e richiede essenzialmente la soluzione di un unico sistema lineare (3.6), di dimensione n_w (la stessa del problema diretto). A confronto, un approccio alle differenze finite, in cui $d\mathbf{w}/d\mathbf{x}$ viene calcolato perturbando una variabile di progetto per volta [24], richiede $n_x + 1$ soluzioni del problema diretto per uno schema del primo ordine e $2n_x$ per uno del secondo ordine. Tale approccio è praticabile solo per spazi di progetto di dimensione ridotta, mentre il metodo aggiunto scala a centinaia o migliaia di variabili senza aumento del costo computazionale.

3.2 Soluzione dell'equazione aggiunta: GMRES e approccio matrix-free

L'equazione aggiunta (3.6) è un sistema lineare della forma

$$\mathbf{A}^T \psi = \mathbf{b}, \quad \mathbf{A} = \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \in \mathbb{R}^{n_w \times n_w}, \quad \mathbf{b} = \left(\frac{\partial f}{\partial \mathbf{w}}\right)^T \in \mathbb{R}^{n_w}. \quad (3.8)$$

La matrice $\mathbf{A}$ è il Jacobiano del residuo rispetto alle variabili di stato, e coincide (trasposto) con la matrice del sistema linearizzato del problema diretto. Essa è in generale *non simmetrica*, *sparsa* e di grandi dimensioni (n_w è dell'ordine del numero di celle della mesh moltiplicato per il numero di equazioni per cella). L'inversione diretta di $\mathbf{A}^T$ è computazionalmente infattibile per sistemi di interesse pratico: si ricorre pertanto a metodi iterativi di tipo Krylov.

3.2.1 Il metodo GMRES

Il metodo GMRES (*Generalized Minimal RESidual*) [25] è il solutore iterativo di riferimento per sistemi lineari non simmetrici di grandi dimensioni. Ad ogni iterazione k , GMRES costruisce la soluzione approssimata ψ_k nel sottospazio di Krylov

$$\mathcal{K}_k(\mathbf{A}^T, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{A}^T \mathbf{b}, (\mathbf{A}^T)^2 \mathbf{b}, \dots, (\mathbf{A}^T)^{k-1} \mathbf{b}\}, \quad (3.9)$$

minimizzando il residuo $\|\mathbf{b} - \mathbf{A}^T \psi_k\|_2$ su tale sottospazio. La costruzione della base ortogonale di $\mathcal{K}_k$ avviene tramite il processo di Arnoldi, che richiede ad ogni passo un solo prodotto matrice-vettore $\mathbf{A}^T \mathbf{v}$. La convergenza è garantita in al più n_w iterazioni (soluzione esatta), ma nella pratica si raggiunge una tolleranza accettabile in un numero di iterazioni molto inferiore, grazie alla struttura spettrale del Jacobiano.

Una caratteristica critica di GMRES è che la memoria richiesta cresce linearmente con il numero di iterazioni, poiché tutti i vettori di Krylov devono essere conservati per

l'ortonormalizzazione. Per controllare l'uso di memoria si utilizza la variante *restarted* GMRES(m), in cui il processo viene riavviato ogni m iterazioni ripartendo dall'ultima soluzione disponibile. La scelta di m è un compromesso tra efficienza di memoria e velocità di convergenza.

3.2.2 Approccio matrix-free

Un aspetto fondamentale dell'implementazione in DAfoam è che il prodotto matrice-vettore $\mathbf{A}^T \mathbf{v}$ richiesto da GMRES non viene mai calcolato assemblando esplicitamente il Jacobiano $\mathbf{A}$. Si adotta invece un approccio *matrix-free* (o *Jacobian-free*) [26], in cui il prodotto è approssimato tramite differenziazione algoritmica (*algorithmic differentiation*, AD) applicata al codice residuo [23]:

$$\mathbf{A}^T \mathbf{v} = \left(\frac{\partial \mathbf{R}}{\partial \mathbf{w}} \right)^T \mathbf{v}, \quad (3.10)$$

calcolato direttamente dal codice sorgente di OpenFOAM senza mai costruire o memorizzare la matrice $\mathbf{A}$. I vantaggi di questo approccio sono molteplici:

- *Risparmio di memoria*: la matrice $\mathbf{A}$ non viene mai formata esplicitamente. Per mesh con milioni di celle, il solo stoccaggio del Jacobiano richiederebbe decine di gigabyte di RAM.
- *Precisione*: la differenziazione algoritmica fornisce derivate esatte al livello della macchina, senza errori di troncamento come nelle differenze finite.
- *Generalità*: lo stesso meccanismo funziona per qualsiasi risolutore OpenFOAM senza richiedere la derivazione manuale del Jacobiano.

3.2.3 Precondizionamento

La velocità di convergenza di GMRES dipende fortemente dalla distribuzione degli autovalori di $\mathbf{A}^T$. Un preconditionatore $\mathbf{P} \approx \mathbf{A}^T$ trasforma il sistema in

$$\mathbf{P}^{-1} \mathbf{A}^T \boldsymbol{\psi} = \mathbf{P}^{-1} \mathbf{b}, \quad (3.11)$$

concentrando lo spettro della matrice preconditionata attorno a 1 e riducendo così il numero di iterazioni necessarie. In DAfoam si utilizzano preconditionatori di tipo ILU (*Incomplete LU factorization*), che approssimano la fattorizzazione LU di $\mathbf{A}^T$ eliminando gli elementi di fill-in al di fuori di una struttura di sparsità fissata. Questa scelta è compatibile con l'approccio matrix-free poiché il preconditionatore può essere costruito su una versione approssimata (colorata) del Jacobiano, mentre il prodotto matrice-vettore esatto resta a carico della AD.

3.3 Calcolo delle derivate parziali

La formula della sensitività (3.7) richiede il calcolo di due ulteriori quantità oltre al vettore aggiunto $\boldsymbol{\psi}$: il termine $\partial f / \partial \mathbf{w}$ (gradiente della funzione obiettivo rispetto alle variabili di stato, necessario anche come termine noto nell'equazione aggiunta) e il termine $\partial \mathbf{R} / \partial \mathbf{x}$ (derivata del residuo rispetto alle variabili di progetto, ossia sensibilità del residuo alle variazioni geometriche).

Il termine $\partial f / \partial \mathbf{w}$ è in genere analitico e di calcolo semplice: per la funzione obiettivo (2.4) definita nel Capitolo 2, il gradiente rispetto alle componenti di velocità è proporzionale all'errore puntuale ai probes e si calcola con poche operazioni algebriche.

Il termine $\partial \mathbf{R} / \partial \mathbf{x}$, invece, cattura come le equazioni discretizzate cambiano al variare della geometria della mesh. Questo termine dipende dalla deformazione della griglia computazionale indotta dallo spostamento dei punti di controllo FFD, ed è calcolato tramite AD applicata alla routine di deformazione della mesh [23]. Il prodotto $\psi^T (\partial \mathbf{R} / \partial \mathbf{x})$ fornisce infine il gradiente cercato rispetto ai punti di controllo, che viene passato all'ottimizzatore per aggiornare la geometria.

Capitolo 4

Metodologia Computazionale e Strumenti

4.1 Architettura del framework e flusso di calcolo

Il framework computazionale utilizzato in questa tesi è costruito interamente su componenti open-source che cooperano in una catena di calcolo automatizzata. L'elemento centrale è **DAFoam** (*Discrete Adjoint with OpenFOAM*) [20], che estende il risolutore CFD OpenFOAM [19] con la capacità di calcolare le sensitività aggiuntive mediante l'approccio dell'aggiunto discreto descritto nel Capitolo 3. DAFoam si interfaccia con l'**ecosistema MACH-Aero**, un framework Python per l'ottimizzazione aerodinamica ad alta fedeltà che integra: la libreria **OpenMDAO** [27] per l'orchestrazione multidisciplinare del ciclo di ottimizzazione, la libreria **PyGeo** per la parametrizzazione geometrica tramite FFD, la libreria **IDWarp** per la deformazione della mesh volumetrica, e la libreria **pyOptSparse** come interfaccia verso gli ottimizzatori gradient-based.

La caratteristica architetturale distintiva di DAFoam è la separazione netta tra una parte *solver-agnostic* che implementa il calcolo delle derivate parziali e la soluzione dell'equazione aggiunta in modo indipendente dal solver specifico, e una parte *solver-specific* in cui lo sviluppatore definisce esclusivamente la forma della funzione residuo e la connettività associata. Questa struttura orientata agli oggetti consente di estendere l'implementazione dell'aggiunto discreto a un nuovo solver OpenFOAM aggiungendo o modificando soltanto poche centinaia di righe di codice sorgente [20]. Nella sua versione attuale, DAFoam fornisce supporto per 8 solver primali, 5 modelli di turbolenza e un modello di radiazione. Le prestazioni del framework sono state validate fino a 10^7 celle su 1536 core CPU in configurazione MPI distribuita, con un errore medio nelle derivate aggiuntive inferiore allo 0.1% rispetto alle differenze finite centrali [20].

Il ciclo di ottimizzazione completo per ogni iterazione si articola nei passi seguenti:

1. L'ottimizzatore aggiorna il vettore delle variabili di progetto $\mathbf{x}$ (coordinate dei punti di controllo FFD).
2. PyGeo propaga la deformazione FFD alle coordinate dei nodi superficiali del corpo.
3. IDWarp deforma la mesh volumetrica in modo coerente con la nuova posizione della superficie, preservando la qualità della griglia.
4. DASimpleFoam risolve le equazioni di Navier–Stokes stazionarie sul dominio deformato, ottenendo il campo di moto $\mathbf{w} = [\mathbf{U}, p]$.
5. Il modulo della funzione obiettivo calcola $f(\mathbf{x}, \mathbf{w})$ confrontando il campo di velocità simulato con quello di riferimento ai probes.
6. Il solutore aggiunto di DAFoam risolve l'equazione (3.6) tramite GMRES, ottenendo il vettore aggiunto ψ .
7. Il gradiente $df/d\mathbf{x}$ viene calcolato secondo (3.7) e restituito all'ottimizzatore.

Il ciclo si ripete fino al soddisfacimento del criterio di convergenza sull'obiettivo o sul gradiente.

4.1.1 Il risolutore DASimpleFoam

Il risolutore CFD impiegato è **DASimpleFoam**, la variante DA Foam del classico **simpleFoam** di OpenFOAM per flussi incomprimibili stazionari. Esso implementa l'algoritmo SIMPLE (*Semi-Implicit Method for Pressure-Linked Equations*) per il disaccoppiamento iterativo del campo di velocità e di pressione. Rispetto al **simpleFoam** standard, DASimpleFoam è strumentato per il calcolo del Jacobiano $\partial \mathbf{R} / \partial \mathbf{w}$ richiesto dall'aggiunto discreto tramite l'approccio *FD Jacobian* [20]: le derivate parziali del residuo rispetto alle variabili di stato sono calcolate con differenze finite accelerate dalla *graph coloring*, una tecnica che raggruppa le perturbazioni di variabili non adiacenti nella topologia della mesh, riducendo il numero di valutazioni del residuo da n_w a poche decine indipendentemente dalla dimensione del problema. Il Jacobiano risultante viene assemblato in formato sparso e utilizzato sia per i prodotti matrice-vettore richiesti da GMRES sia come base per il preconditionatore ILU. Nella configurazione adottata, la convergenza del problema primale è imposta ad una tolleranza molto stringente ($\|\mathbf{R}\|_\infty < 10^{-13}$) per garantire che il punto di partenza per la soluzione aggiunta sia sufficientemente vicino alla soluzione esatta; ciò è necessario per l'accuratezza delle sensitività.

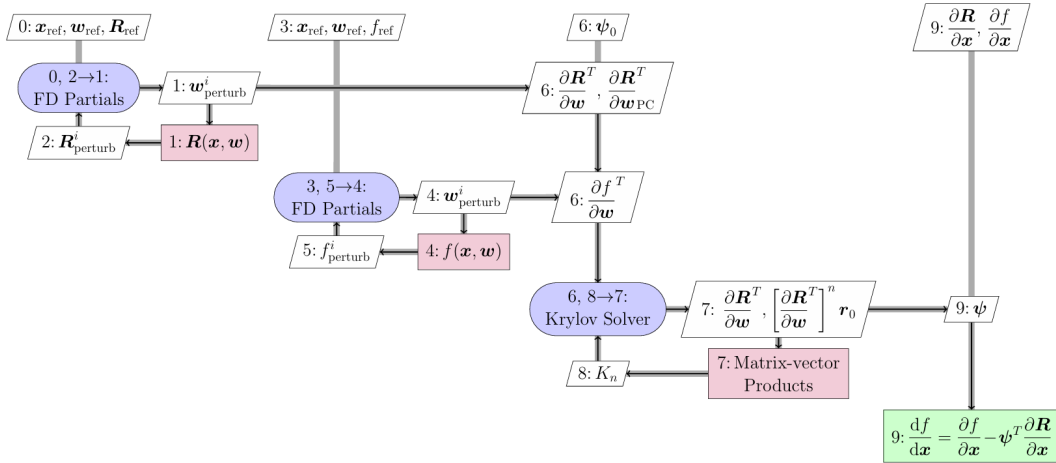


Figura 4.1: Flusso di dati e di processo per l'approccio FD Jacobian solver-agnostic in DA Foam [20]. I primi due blocchi viola rappresentano i calcoli delle derivate parziali tramite differenze finite (step 0–2 per $\partial \mathbf{R}^T / \partial \mathbf{w}$ e step 3–5 per $\partial f^T / \partial \mathbf{w}$); il terzo blocco viola è il solutore di Krylov (GMRES, step 6–8) che produce il vettore aggiunto ψ (step 9); il blocco verde è la formula finale della sensitività $df/d\mathbf{x} = \partial f / \partial \mathbf{x} - \psi^T \partial \mathbf{R} / \partial \mathbf{x}$.

4.1.2 Gli ottimizzatori: SLSQP e IPOPT

Nel presente lavoro si utilizzano due ottimizzatori gradient-based, accessibili tramite la libreria **pyOptSparse**:

SLSQP (*Sequential Least Squares Quadratic Programming*) [28] è un metodo di ottimizzazione vincolata che approssima iterativamente il problema originale con una sequenza di sottoproblemi di programmazione quadratica. È particolarmente efficiente per problemi di piccola e media dimensione e non richiede la fattorizzazione di grandi sistemi lineari ad ogni iterazione.

IPOPT (*Interior Point OPTimizer*) [29] è un metodo a punto interno di ordine superiore, progettato per problemi di grandi dimensioni con vincoli non lineari. Utilizza il metodo di Newton per risolvere le condizioni di ottimalità KKT (*Karush–Kuhn–Tucker*) e gestisce internamente i vincoli di non negatività tramite funzioni barriera logaritmiche. In questa tesi, entrambi gli ottimizzatori vengono confrontati sul caso di riferimento (ellisse → cilindro) per valutarne la robustezza e l’efficienza di convergenza.

Nota sulla scelta della funzione obiettivo. Esperimenti preliminari condotti con l’ellisse di semi-asse minore $b = 0.9$ (geometria di partenza più vicina al cilindro target) hanno confrontato la formulazione **Variance** classica, in cui il denominatore di normalizzazione è la varianza globale del campo di riferimento, con la formulazione **VarianceScaled** adottata in questa tesi, che normalizza puntualmente per il valore locale di riferimento (si veda la Sezione 2.2.2). I risultati mostrano che **VarianceScaled** converge in modo leggermente migliore nella scia lontana ($x \in [6R, 9R]$), mentre nella scia intermedia ($x \in [3R, 6R]$) le due formulazioni producono risultati comparabili. Sulla base di questi test preliminari, **VarianceScaled** è stata adottata come formulazione di riferimento per tutti i casi con $b = 0.7$. Il confronto diretto tra le due formulazioni a parità di configurazione di probes viene ripreso nel Capitolo 6.

4.2 Free Form Deformation

La *Free Form Deformation* (FFD) [18] è la tecnica adottata per parametrizzare la geometria del corpo e definire lo spazio delle variabili di progetto. L’idea fondamentale è quella di incorporare il corpo in un volume di controllo reticolare (*FFD box*), mappare le coordinate dei nodi della mesh superficiale nel sistema di riferimento del box tramite una trasformazione isoparametrica, e deformare la geometria spostando i punti di controllo del reticolo.

4.2.1 Il box FFD per il cilindro

Per il caso cilindro, il box FFD è un parallelepipedo rettangolare con:

- **Dimensioni:** $x \in [-1, 1]$, $y \in [-1.25, 1.25]$, $z \in [0, 1]$, centrato sull’asse del cilindro (raggio $R = 1$, diametro $D = 2$); il box si estende fino a $1.25 R$ in direzione y , garantendo che l’intero corpo sia contenuto anche durante le deformazioni intermedie;
- **Griglia di controllo:** $5 \times 2 \times 2$ punti, dove 5 è il numero di stazioni lungo x (in $x \in \{-1, -0.5, 0, 0.5, 1\}$), 2 lungo y (riga superiore $y = +1$ e inferiore $y = -1$) e 2 lungo z (le due facce della slice 2D).

In totale, il box conta $5 \times 2 \times 2 = 20$ punti di controllo. La deformazione è vincolata ad essere *simmetrica* rispetto all’asse x , in modo da preservare la simmetria superiore-inferiore del corpo: per ogni stazione i la riga superiore si sposta di $+\delta_i$ e quella inferiore di $-\delta_i$, gonfiando o sgonfiando il profilo localmente. Il numero di variabili di progetto indipendenti è quindi 5 (una per ciascuna delle 5 stazioni x , incluse le stazioni di leading e trailing edge, lasciate libere per consentire l’arrotondamento caratteristico del cilindro).

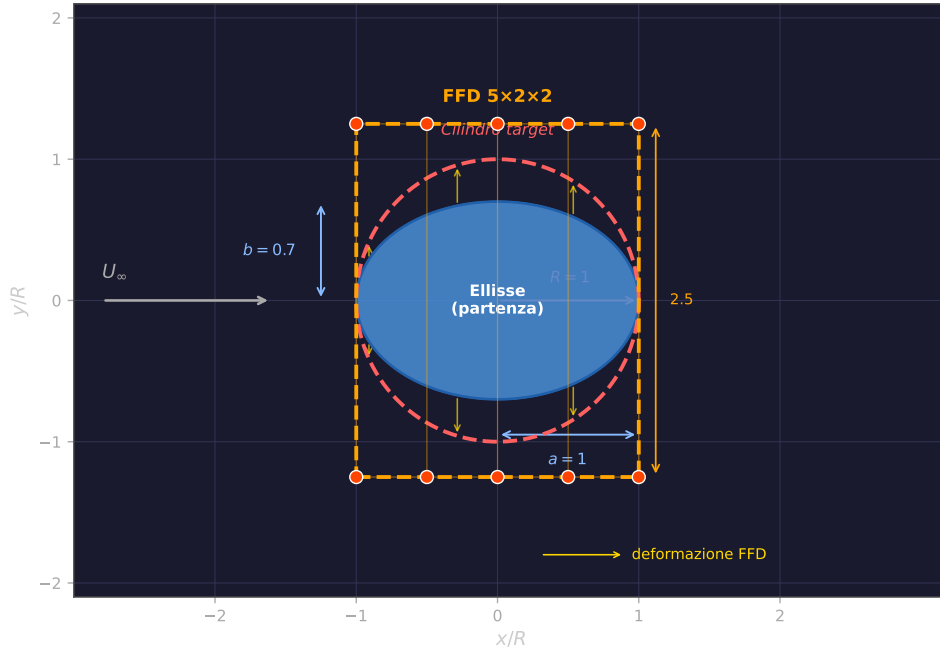


Figura 4.2: Schema del box FFD per il caso cilindro. In blu: ellisse di partenza ($a = 1$, $b = 0.7$). In rosso tratteggiato: cilindro target ($R = 1$). In arancio tratteggiato: box FFD $5 \times 2 \times 2$ ($x \in [-R, R]$, $y \in [-1.25R, +1.25R]$); i punti rossi sono i 10 punti di controllo nel piano $z = 0$. Le frecce gialle indicano la direzione della deformazione dalla geometria iniziale verso quella target.

4.2.2 Propagazione della deformazione alla mesh

Una volta aggiornate le coordinate dei punti di controllo FFD, la deformazione viene propagata in tre passi:

1. **Deformazione superficiale** (PyGeo): le nuove coordinate dei nodi della superficie del corpo vengono calcolate dalla mappa isoparametrica del box FFD.
2. **Deformazione volumetrica** (IDWarp): il moto dei nodi superficiali viene esteso a tutto il dominio fluido tramite la libreria IDWarp (*Inverse Distance Weighting mesh deformation*), che calcola lo spostamento di ogni nodo interno come media pesata degli spostamenti dei nodi superficiali, con pesi decrescenti con la distanza dalla superficie. Questo garantisce che la mesh rimanga di buona qualità anche per deformazioni significative.
3. **Ricalcolo della geometria di cella**: OpenFOAM aggiorna i volumi, le aree e i centri di cella per la nuova configurazione della mesh.

4.3 Setup dell'ambiente di calcolo

4.3.1 Containerizzazione con Docker

L'intero stack software [OpenFOAM, DAfoam, MACH-Aero, MPI e le relative dipendenze] è distribuito come immagine **Docker**, un container Linux che incapsula l'ambiente

di esecuzione in modo riproducibile e portabile. L'uso di Docker elimina i problemi di compatibilità tra versioni di librerie e sistemi operativi diversi, consentendo di eseguire le stesse simulazioni senza modifiche su macchine diverse (workstation locale, cluster di calcolo, cloud). La compilazione dei sorgenti OpenFOAM e DAFoam all'interno del container è gestita dallo script `Allwmake`, che compila tutti i moduli nelle versioni specifiche richieste dal framework.

4.3.2 Parallelizzazione MPI

Le simulazioni sono eseguite in parallelo su $N_{\text{proc}} = 8$ processi MPI, con la decomposizione del dominio gestita da OpenFOAM tramite il metodo `scotch` (graph partitioning). La libreria **mpi4py** fornisce l'interfaccia Python per la comunicazione tra processi nei moduli OpenMDAO e DAFoam. Il vettore aggiunto ψ e il prodotto matrice-vettore $(\partial \mathbf{R} / \partial \mathbf{w})^T \mathbf{v}$ richiesti da GMRES vengono calcolati in parallelo distribuito, con le operazioni di reduce e scatter gestite internamente da DAFoam. I parametri del solutore GMRES adottati sono: tolleranza relativa 10^{-14} , massimo 5000 iterazioni, restart ogni 150 iterazioni, preconditionatore ILU con fill level 1 e riordinamento RCM (*Reverse Cuthill–McKee*).

4.3.3 Script di automazione

Il ciclo di ottimizzazione è interamente automatizzato dallo script Python `runScript.py`, che coordina tutti i componenti del framework tramite OpenMDAO. Lo script definisce il modello multidisciplinare come grafo di componenti connessi (`Multipoint` con `ScenarioAerodynamic`), imposta le variabili di progetto, la funzione obiettivo e gli eventuali vincoli, e lancia l'ottimizzatore. I risultati [storia della funzione obiettivo, storia dei residui, coordinate della forma ottimizzata] vengono salvati automaticamente nella directory del caso durante l'esecuzione.

Capitolo 5

Impostazione del Caso di Studio

Questo capitolo descrive in dettaglio la configurazione pratica del caso di inverse design sul corpo tozzo, traducendo il framework teorico dei Capitoli 2–4 in una sequenza concreta di file di input e di script eseguibili. Il caso di riferimento (*testgenerale*) considera la deformazione di un'ellisse con semi-assi $a = 1$ e $b = 0.7$ verso il cilindro circolare di raggio $R = 1$, in flusso laminare a $Re = 40$, con le componenti di velocità campionate da 120 probes disposti in una griglia rettangolare nella regione di scia.

5.1 Struttura generale del caso DA Foam e automazione tramite runScript.py

Il ciclo di ottimizzazione è interamente orchestrato dallo script `runScript.py`, scritto in Python e basato su OpenMDAO. Lo script definisce un modello multidisciplinare attraverso la classe `Top`, che eredita da `Multipoint` (l'elemento radice di MACH-Aero per la gestione di scenari multipli) e assembla il grafo di calcolo schematizzato di seguito.

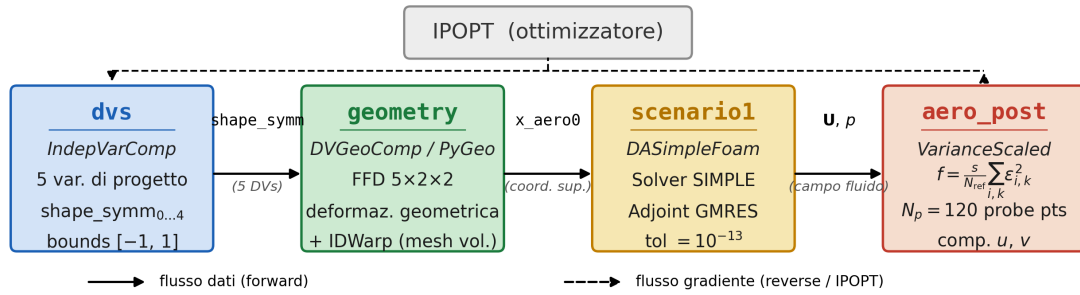


Figura 5.1: Grafo di calcolo OpenMDAO: flusso dati in avanti (freccie piene) e flusso del gradiente verso IPOPT (freccie tratteggiate).

Il grafo è composto dai seguenti sottosistemi:

dvs (IndepVarComp). Componente OpenMDAO che ospita e promuove le variabili di progetto indipendenti. In questo caso espone due variabili: `shape_symm` (vettore di 5 componenti, una per ciascuna stazione x della FFD) e `patchV` (vettore 2D che codifica intensità e angolo del flusso di ingresso, fissati a $U_\infty = 1$ e $\alpha = 0$).

mesh (DAFoamBuilder.get_mesh_coordinate_subsystem). Legge le coordinate superficiali iniziali della mesh OpenFOAM ($N_s = 460$ nodi sulla superficie dell'ellisse) e le espone come variabile `x_aero0`, che costituisce il punto di partenza per la deformazione geometrica.

geometry (OM.DVGEOCOMP). Wrappa la libreria PyGeo e carica il file di controllo FFD `FFD/cylinderFFD.xyz`. Nel metodo `configure()` si registrano le *shape functions* simmetriche: per ogni stazione $i \in \{0, \dots, 4\}$ corrispondente alle x -positions

$\{-1, -0.5, 0, 0.5, 1\}$, si definisce una coppia di spostamenti $+\delta_i$ (fila superiore, $j = 1$) e $-\delta_i$ (fila inferiore, $j = 0$), ottenendo una deformazione simmetrica rispetto all'asse x che conserva la simmetria superiore-inferiore del corpo. Le 5 shape functions vengono raggruppate nel design variable `shape_symm`, con limiti $\delta_i \in [-1, 1]$ (in unità di R).

scenario1 (ScenarioAerodynamic). Contiene il risolutore `DASimpleFoam` e il solutore aggiunto. Riceve in ingresso le coordinate aggiornate della superficie `x_aero` (output di `geometry`) e la velocità di ingresso `patchV`; produce in uscita il campo di moto $[U, p]$ e il valore dell'obiettivo f tramite il sottocomponente `aero_post`. La connessione `mesh.x_aero0 → geometry.x_aero_in` e `geometry.x_aero0 → scenario1.x_aero` è stabilita manualmente nel metodo `setup()`, seguendo la convenzione MACH-Aero. Il metodo `configure()`, invocato da `OpenMDAO` dopo la fase di `setup()`, completa la registrazione dei design variables sull'ottimizzatore:

- `add_design_var("shape_symm", lower=-1, upper=1, scaler=1.0)`: le 5 variabili simmetriche con limiti box.
- `add_objective("scenario1.aero_post.Var", scaler=10.0)`: l'obiettivo viene moltiplicato per 10 prima di essere passato all'ottimizzatore, per normalizzarne l'ordine di grandezza.

Lancio e modalità operative. Lo script accetta due argomenti da riga di comando:

- `-optimizer`: sceglie tra IPOPT (predefinito), SLSQP e SNOPT.
- `-task`: seleziona la modalità di esecuzione. Con `run_driver` si avvia l'ottimizzazione completa; `run_model` esegue una singola valutazione del primale (utile per verificare la mesh e le condizioni al contorno); `check_totals` confronta le sensitività aggiuntive con la differenza finita centrale (passo $\Delta x = 10^{-3}$), consentendo la validazione del Jacobiano prima dell'ottimizzazione.

Al termine di ogni ottimizzazione, i risultati vengono salvati automaticamente:

- `OptView.hst`: file binario in formato `pyOptSparse` con la storia completa dell'ottimizzazione (variabili, obiettivo, gradienti ad ogni iterazione), leggibile con il tool grafico `OptView`.
- `opt_IPOPT.txt`: log testuale di IPOPT con le colonne iterazione, obiettivo, `inf_pr`, `inf_du`, $\lg(\mu)$, $\|d\|$, α per ogni iterazione.
- `shape_state.txt`: vettore finale delle variabili di progetto δ^* , utile per riprendere l'ottimizzazione o per analisi post-processing.
- `objective_history.csv` e `objective_plot.png`: storia dell'obiettivo in formato CSV e il relativo grafico, generati dal blocco di post-processing integrato nello script.

5.2 Generazione e deformazione della mesh: configurazione

5.2.1 Geometria iniziale e target

Il corpo iniziale del caso *testgenerale* è un'ellisse con semi-asse maggiore $a = 1$ nella direzione x (direzione del flusso) e semi-asse minore $b = 0.7$ nella direzione y (direzione trasversale). Il semi-asse maggiore coincide con il raggio del cilindro target ($a = R = 1$), quindi entrambi i corpi condividono la stessa estensione orizzontale $[-R, R] = [-1, 1]$ e la stessa lunghezza caratteristica $D = 2R = 2$ usata nel calcolo del numero di Reynolds. La differenza è esclusivamente di forma: l'ellisse è più piatta nella direzione normale al flusso ($b = 0.7 < R = 1$), e l'obiettivo dell'ottimizzazione inversa è deformarla fino a riprodurre la sezione circolare del cilindro, modificando i parametri geometrici FFD senza alterarne la scala. La geometria **target** è il **cilindro circolare** di raggio $R = 1$, la cui soluzione di flusso laminare stazionario a $Re = 40$ costituisce il campo di velocità di riferimento $\mathbf{U}^{\text{ref}}$.

Per produrre il campo di riferimento, si esegue una singola simulazione CFD sulla mesh del cilindro con `-task run.model`. Il campo di velocità risultante viene scritto come file `UData` nella directory del caso e letto ad ogni iterazione dal modulo `varianceScaled` per calcolare l'errore relativo ai probes. Il campo di riferimento è interpolato alle posizioni dei probes tramite il meccanismo interno di `DAFoam`, che utilizza l'interpolazione bilineare sulla mesh poliedrica di `OpenFOAM`. La Figura 5.2 mostra il campo di velocità stazionario attorno all'ellisse iniziale a $Re = 40$: si osserva la scia simmetrica con la bolla di ricircolazione e le linee di flusso che si richiudono a valle del corpo.

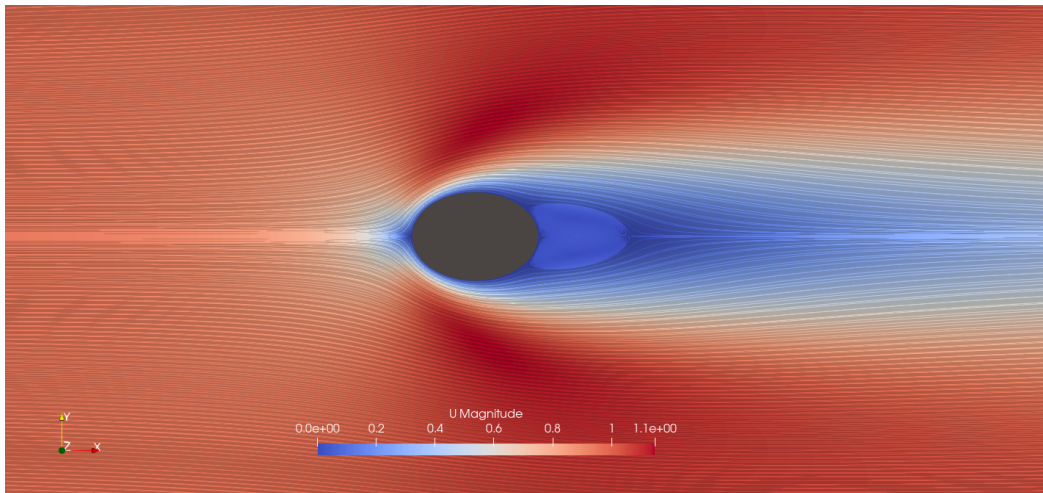


Figura 5.2: Campo di velocità (magnitudo $|\mathbf{U}|$) con linee di flusso attorno all'ellisse di partenza ($a = 1$, $b = 0.7$) a $Re = 40$. Il rosso indica alta velocità (zona di accelerazione laterale), il blu la regione di scia a velocità ridotta. La bolla di ricircolazione simmetrica è visibile immediatamente a valle del corpo.

5.2.2 Topologia e parametri della mesh

La mesh computazionale è una griglia strutturata di tipo **O-mesh**, generata tramite il dizionario `blockMeshDict` di `OpenFOAM` e composta da 4 blocchi quadrilaterali disposti attorno al corpo.

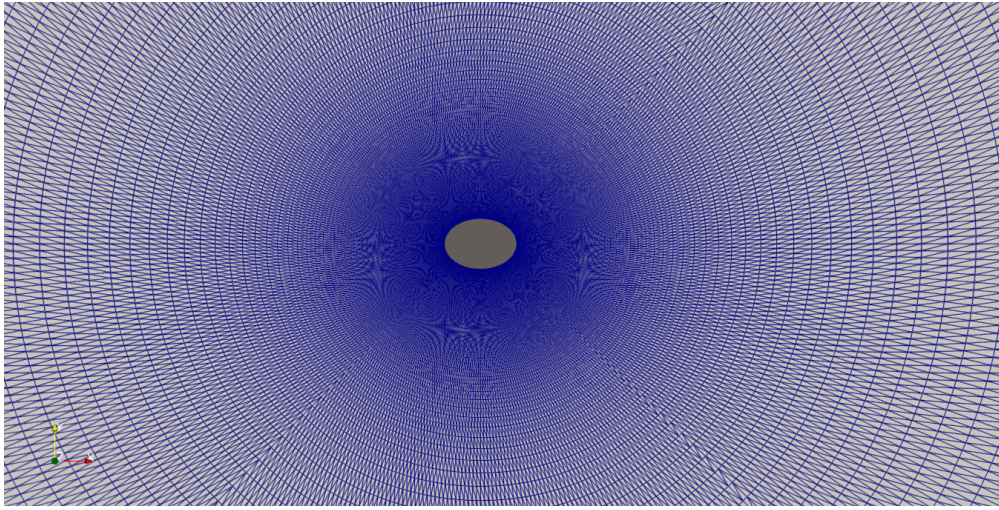


Figura 5.3: Vista globale della O-mesh strutturata ($100 \times 100 \times 1$ celle per blocco, 4 blocchi, dominio circolare $R_\infty = 30R$). I quattro blocchi quadrilaterali attorno all'ellisse sono chiaramente visibili; il forte infittimento verso la parete (grading 1/500) non è apprezzabile a questa scala.

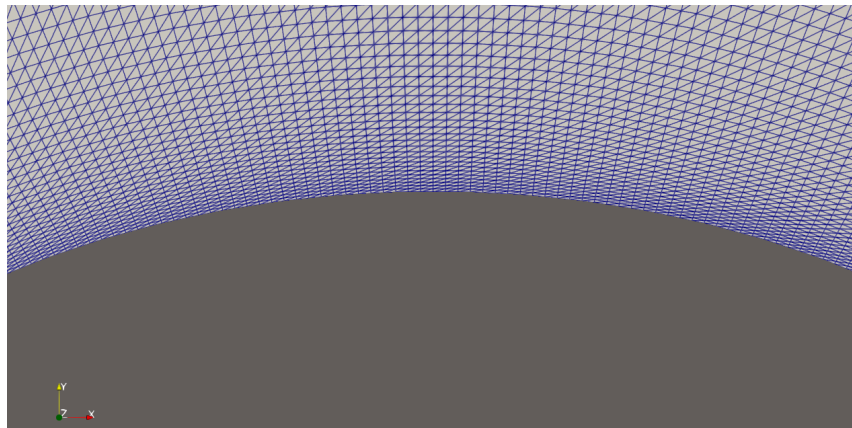


Figura 5.4: Zoom della mesh nella regione di parete. Il raffinamento progressivo (grading 1/500 nella direzione normale) garantisce la risoluzione dello strato limite laminare a $Re = 40$; il rapporto di aspetto massimo delle celle a parete è ≈ 300 .

- **Contorno interno:** l'ellisse con $a = 1$, $b = 0.7$, discretizzata con bordi polyLine che passano esattamente per i punti sull'ellisse $(\cos \theta, 0.7 \sin \theta)$ per 20 punti intermedi per quadrante.
- **Contorno esterno:** un cerchio di raggio $R_\infty = 30 R$, discretizzato con bordi arc passanti per i mezzipunti a $0^\circ, 90^\circ, 180^\circ, 270^\circ$.
- **Quattro blocchi:** ognuno composto da $100 \times 100 \times 1$ celle, per un totale di $N_{\text{celle}} = 40\,000$ celle. La seconda dimensione (100 celle) è nella direzione normale alla parete, con raffinamento ottenuto tramite simpleGrading con rapporto 1/500 (le celle vicino alla parete sono 500 volte più piccole di quelle esterne). La prima dimensione (100 celle) è nella direzione tangenziale alla parete, con grading uniforme ($= 1$) per i blocchi inferiore e superiore, e grading 500 e 1/500 rispettivamente per il blocco destro (lato di trailing edge) e sinistro (lato di leading edge).

I parametri fisici della simulazione sono riassunti nella Tabella 5.1; la qualità della mesh, verificata con l'utilità checkMesh di OpenFOAM, è riportata nella Tabella 5.2.

Tabella 5.1: Parametri fisici e numerici del caso test generale.

Parametro	Simbolo	Valore
Velocità di freestream	U_∞	1.0 m/s
Viscosità cinematica	ν	0.05 m ² /s
Numero di Reynolds	$Re = U_\infty D / \nu$	40
Raggio del cilindro target	R	1.0 m
Raggio far-field	R_∞	30.0 m
Numero totale di celle	N_{celle}	40 000
Tolleranza primale	$\ \mathbf{R}\ _\infty$	10^{-13}
Iterazioni massime SIMPLE	—	10 000

Tabella 5.2: Parametri di qualità della mesh (output checkMesh).

Parametro	Valore	Soglia accettabile
Dominio computazionale	$[-30, 30] \times [-30, 30] \times [0, 1]$	—
Non-ortogonalità massima	12.0	< 70
Non-ortogonalità media	4.3	—
Skewness massima	0.128	< 4
Aspect ratio massimo	300.6	< 1000
Area minima di faccia	$4.16 \times 10^{-5} \text{ m}^2$	—
Area massima di faccia	1.786 m^2	—
Volume minimo di cella	$4.16 \times 10^{-5} \text{ m}^3$	> 0
Volume massimo di cella	0.817 m^3	—
Volume totale del dominio	2825.1 m^3	—

I parametri di qualità indicano una mesh eccellente: la non-ortogonalità massima di 12 è ben al di sotto della soglia critica (70) e la skewness di 0.13 è trascurabile. L'aspect ratio

elevato (≈ 300) è atteso e non problematico: deriva necessariamente dal raffinamento parietale con `simpleGrading` (rapporto $1/500$), che produce celle molto allungate vicino alla parete per risolvere correttamente i gradienti normali di velocità. Il rapporto tra volume massimo e minimo di cella ($\approx 1.97 \times 10^4$) riflette la stessa gerarchia di raffinamento.

5.2.3 Condizioni al contorno

Le condizioni al contorno sono specificate tramite le patch di OpenFOAM:

- **Superficie del corpo** (cylinder, tipo wall): condizione di no-slip, $\mathbf{U} = \mathbf{0}$; pressione: gradiente nullo normale alla parete.
- **Far-field** (farField, tipo patch): velocità uniforme $\mathbf{U} = (U_\infty, 0, 0)$ in ingresso; pressione cinematica nulla, $p = 0$. OpenFOAM gestisce automaticamente la distinzione ingresso/uscita tramite il segno del flusso normale.
- **Facce frontale e posteriore** (backAndFront, tipo symmetry): condizione di simmetria nelle due direzioni z , che riduce il problema 3D a una slice 2D con spessore $\Delta z = 1 R$.

5.2.4 Schema di soluzione e parametri numerici SIMPLE

Il risolutore DASSimpleFoam implementa l'algoritmo SIMPLE con i seguenti parametri (fvSolution):

- **Pressione**: solutore GAMG con smoother Gauss-Seidel, tolleranza relativa 10^{-1} .
- **Velocità**: solutore smoothSolver con smoother Gauss-Seidel, tolleranza relativa 10^{-1} , 1 sweep per iterazione.
- **Fattori di rilassamento**: $\alpha_p = 0.30$ per la pressione, $\alpha_U = 0.70$ per la velocità. Questi valori conservativi garantiscono la convergenza stabile a $Re = 40$.

La tolleranza di arresto del primale è $\|\mathbf{R}\|_\infty < 10^{-13}$, valore molto più stringente di quanto necessario per la sola soluzione CFD, ma indispensabile per garantire la precisione delle sensitività aggiuntive (cfr. Capitolo 3).

5.2.5 Deformazione della mesh con IDWarp

Ad ogni iterazione dell'ottimizzatore, quando la FFD sposta i nodi superficiali dell'ellisse verso la nuova configurazione geometrica, la deformazione deve essere propagata a tutta la mesh volumetrica. IDWarp è configurato con:

- `useRotations = True`: IDWarp utilizza anche le componenti rotazionali dello spostamento superficiale, migliorando la qualità della mesh deformata per grandi spostamenti.
- `evalMode = "exact"`: il campo di deformazione viene calcolato in modo esatto (non approssimato), al costo di un maggior tempo di calcolo.
- `symmetryPlanes`: due piani di simmetria a $z = 0$ e $z = 1$, coerenti con la geometria 2D.

5.3 Implementazione della Funzione Obiettivo

5.3.1 Preparazione del campo di riferimento

Prima di avviare l'ottimizzazione, è necessario generare il campo di velocità di riferimento $\mathbf{U}^{\text{ref}}$ attorno alla geometria target (cilindro circolare). Questo è ottenuto con una singola simulazione CFD sulla mesh del cilindro, eseguita con lo stesso risolutore DASimpleFoam e le stesse condizioni al contorno del caso di ottimizzazione. Il campo risultante viene scritto in `0.0001/U` (al tempo corrispondente alla convergenza del SIMPLE) e copiato nella variabile `UData` tramite lo script `mapping.py`, che mappa il campo di velocità dalla mesh del cilindro alla mesh dell'ellisse mediante interpolazione con `mapFields` di OpenFOAM. In questo modo, il vettore `UData` è disponibile ad ogni cella della mesh dell'ellisse con il valore di $\mathbf{U}^{\text{ref}}$ interpolato nel centroide di cella, e il modulo `DAFoam` lo interpola ulteriormente alle posizioni dei probes.

5.3.2 Approccio preliminare: modulo della differenza vettoriale

Un primo tentativo di funzione obiettivo utilizza direttamente la norma della differenza vettoriale di velocità:

$$f_{\text{naive}}(\mathbf{x}, \mathbf{w}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\mathbf{U}(\mathbf{x}_i) - \mathbf{U}^{\text{ref}}(\mathbf{x}_i)\|^2, \quad (5.1)$$

che somma i quadrati degli scarti vettoriali alle posizioni dei probes e normalizza per il numero di probes. Questo approccio ha il vantaggio della semplicità e corrisponde all'errore quadratico medio (MSE) sui valori di velocità ai probes.

Tuttavia, tale formulazione presenta un limite critico: la contribuzione di ciascun probe alla funzione obiettivo è proporzionale al quadrato della velocità assoluta in quel punto, non all'errore relativo. Nelle regioni dove la velocità di riferimento è prossima a zero (in particolare nella scia immediata del corpo, dove le componenti di velocità possono scendere a valori molto piccoli) anche errori relativamente grandi in termini percentuali contribuiscono in modo trascurabile alla funzione obiettivo, mentre gli errori nelle regioni ad alta velocità dominano. Questa disparità di peso può portare l'ottimizzatore a correggere preferenzialmente la forma nelle zone di alta velocità, trascurando la struttura della scia. Un secondo problema riguarda la dipendenza dall'intensità del flusso: se le unità o la scala della velocità vengono modificate (ad es. cambiando U_∞), il valore di f_{naive} scala con U_∞^2 , rendendo i confronti tra configurazioni non immediati.

5.3.3 Implementazione della funzione `VarianceScaled`

Per superare le limitazioni descritte, si adotta la funzione obiettivo `VarianceScaled`, implementata nel modulo C++ `DAFunctionVarianceScaled` di `DAFoam`. Il nucleo del calcolo, estratto dal codice sorgente, è il seguente:

```
forAll(probePoints, ptI)
{
    // interpola U corrente e U di riferimento al probe ptI
    vector Usim = interpolate(U,      probePoints[ptI]);
    vector Uref = interpolate(UData, probePoints[ptI]);
```

```

forAll(components, compI)
{
    label k = components[compI];
    scalar refVal = Uref[k];
    scalar varDif = (Usim[k] - refVal) / refVal;    // errore relativo
    functionValue += scale * varDif * varDif;
    nRefPoints++;
}
}
functionValue /= nRefPoints;

```

Il ciclo esterno scorre tutti gli N_p probes; per ognuno interpola il campo di velocità simulato $\mathbf{U}$ e il campo di riferimento $\mathbf{U}^{\text{ref}}$ (letto da `UData`) alla posizione del probe tramite interpolazione bilineare sulla mesh. Il ciclo interno scorre le componenti selezionate ($k \in \{0, 1\}$ per u e v): per ciascuna calcola l'errore relativo puntuale

$$\epsilon_{i,k} = \frac{U_k(\mathbf{x}_i) - U_k^{\text{ref}}(\mathbf{x}_i)}{U_k^{\text{ref}}(\mathbf{x}_i)}, \quad (5.2)$$

ne somma il quadrato moltiplicato per il fattore di scala s , e conta il numero totale di termini $N_{\text{ref}} = N_p \cdot |K|$. Al termine del doppio ciclo la funzione obiettivo è

$$f = \frac{s}{N_{\text{ref}}} \sum_{i=1}^{N_p} \sum_{k \in K} \epsilon_{i,k}^2, \quad (5.3)$$

che coincide con la formula (2.4) del Capitolo 2. La divisione per N_{ref} rende f indipendente dal numero di probes, consentendo confronti coerenti tra configurazioni diverse.

Il termine `addToAdjoint`: `True` indica a `DAFoam` di calcolare il gradiente $\partial f / \partial \mathbf{U}$ necessario come termine noto dell'equazione aggiunta (3.6). Derivando rispetto a $U_k(\mathbf{x}_i)$:

$$\frac{\partial f}{\partial U_k(\mathbf{x}_i)} = \frac{2s}{N_{\text{ref}}} \cdot \frac{U_k(\mathbf{x}_i) - U_k^{\text{ref}}(\mathbf{x}_i)}{[U_k^{\text{ref}}(\mathbf{x}_i)]^2}. \quad (5.4)$$

Questo vettore-seme, calcolato analiticamente da `DAFoam`, viene usato per inizializzare la soluzione GMRES del sistema aggiunto, garantendo la precisione delle sensitività al livello della macchina.

La configurazione nel `runScript.py` è la seguente:

```

"Var": {
    "type": "varianceScaled",
    "source": "boxToCell",
    "min": [-20, -20, -10], "max": [20, 20, 10],
    "scale": 10.0,
    "mode": "probePoint",
    "probePointCoords": probePointCoords["probePointCoords"],
    "varName": "U",

```

```

    "data":          "UData",
    "varType":       "vector",
    "components":    [0, 1],
    "timeDependentRefData": False,
    "addToAdjoint":  True,
}

```

Il parametro `scale: 10.0` riporta f_0 all'ordine di 10^4 – 10^5 per quasi tutti i tests, un intervallo ben condizionato per IPOPT (tranne quelli calcolati in scia lontana e agli estremi del dominio, che stanno intorno a 10^0 – 10^2 , valori comunque più che accettabili). Il campo di riferimento `UData` è stazionario (`timeDependentRefData: False`) e viene caricato una sola volta prima dell'ottimizzazione.

5.3.4 Configurazione generale dei probes e varianti

La griglia di probes è generata parametricamente tramite uno script Matlab dedicato, che riceve in input le coordinate di inizio e fine del dominio in x , il range in y e il numero di stazioni per direzione, e produce il file `probePointCoords.json`. Per tutti i casi sul cilindro, il range trasversale è fissato a $y \in [-1.5R, 1.5R]$ (tre raggi in totale), con 15 stazioni uniformi (passo $\Delta y \approx 0.214 R$).

Il punto di partenza dello studio parametrico è una **griglia piena** $15 \times 15 = 225$ **probes** che copre una finestra di $3R \times 3R$ nella scia. Tre intervalli in x vengono esplorati sistematicamente (Capitolo 6):

- **Scia prossima** (Probes15_45x3): $x \in [1.5R, 4.5R]$, passo $\Delta x \approx 0.214 R$. È la regione immediatamente a valle del corpo, dove la differenza geometrica tra ellisse e cilindro produce il contrasto di scia più intenso.
- **Scia intermedia** (Probes3_6x3): $x \in [3.0R, 6.0R]$, passo $\Delta x \approx 0.214 R$. La bolla di ricircolazione si è già chiusa e la scia è più regolare.
- **Scia lontana** (Probes6_9x3): $x \in [6.0R, 9.0R]$, passo $\Delta x \approx 0.214 R$, con la consueta ellisse di partenza ($a = 1$, $b = 0.7$). A questa distanza le differenze tra i campi si attenuano; questo caso serve a quantificare la perdita di informazione geometrica con la distanza.

Per ciascun intervallo si studiano poi due varianti ridotte, ottenute rimuovendo progressivamente le stazioni più distanti dal corpo:

- **Griglia $8 \times 15 = 120$ probes**: si mantengono le prime 8 stazioni in x (metà frontale della finestra). Per la scia prossima, ad esempio, si passa da $x \in [1.5R, 4.5R]$ a $x \in [1.5R, 3.0R]$ (testgenerale, caso di riferimento per il Capitolo 7).
- **Griglia $4 \times 15 = 60$ probes**: si mantengono le prime 4 stazioni in x (primo quarto della finestra), spingendo ulteriormente la misura verso il corpo.

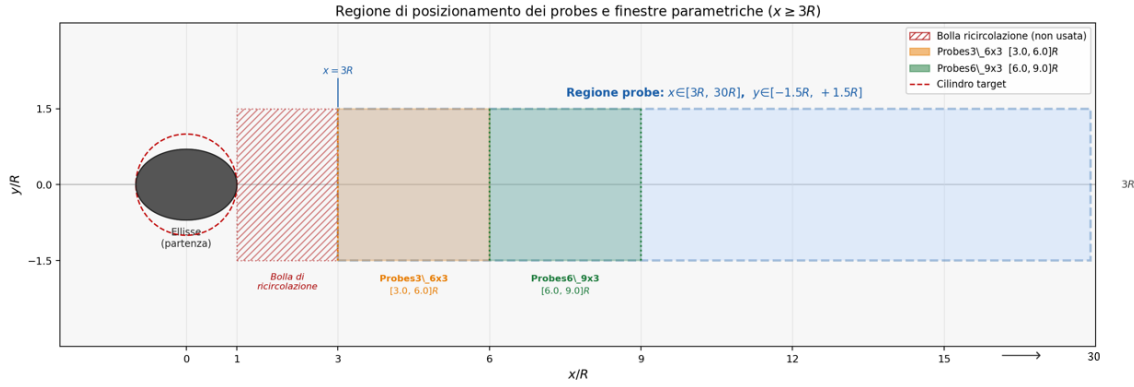


Figura 5.5: Regione generale di posizionamento dei probes ($x \in [3R, 30R]$, $y \in [-1.5R, +1.5R]$, tratteggio blu). La zona a tratteggio rosso ($x \in [1R, 3R]$) indica la zona della bolla di ricircolazione, punto critico dell'analisi generale per l'instabilità dell'errore relativo. Le due finestre principali dello studio parametrico: scia intermedia *Probes3_6x3* (arancio, $[3R, 6R]$) e scia lontana *Probes6_9x3* (verde, $[6R, 9R]$). L'elisse (profilo di partenza, grigio) e il cilindro target (rosso tratteggiato) sono mostrati in scala.

La scelta di non includere probes nella regione $x < 3R$ è motivata dalla presenza della bolla di ricircolazione di scia: a valle immediato del corpo, le componenti di velocità possono passare per zero o cambiare segno, rendendo l'errore relativo $\epsilon_{i,k} = (U_k - U_k^{\text{ref}})/U_k^{\text{ref}}$ numericamente instabile. La Figura 5.6 mostra la struttura della bolla nel campo di riferimento del cilindro a $Re = 40$: i due vortici simmetrici controrotanti sono confinati entro $x \lesssim 3R$ dalla superficie. La coordinata $z = 0.05R$ è scelta all'interno della slice 2D (spessore $\Delta z = 1R$) per evitare le facce di simmetria.

5.3.5 Ottimizzatore IPOPT: configurazione

L'ottimizzatore scelto per il caso *testgenerale* è IPOPT (versione 3.13.5) con solutore lineare MUMPS. I parametri principali impostati in `runScript.py` sono:

- `tol`: `1e-6`: tolleranza sull'ottimalità KKT complessiva (condizione di arresto principale).
- `constr_viol_tol`: `1e-7`: tolleranza sulla violazione del vincolo di box sulle variabili di progetto.
- `max_iter`: `200`: numero massimo di iterazioni consentite.
- `mu_strategy`: `"adaptive"`: il parametro di barriera μ viene aggiornato in modo adattivo, migliorando la convergenza rispetto alla strategia monotona.
- `limited_memory_max_history`: `30`: approssimazione L-BFGS della Hessiana con finestra di storia di 30 iterazioni.
- `nlp_scaling_method`: `"gradient-based"`: scalatura automatica delle variabili basata sui gradienti al punto iniziale.

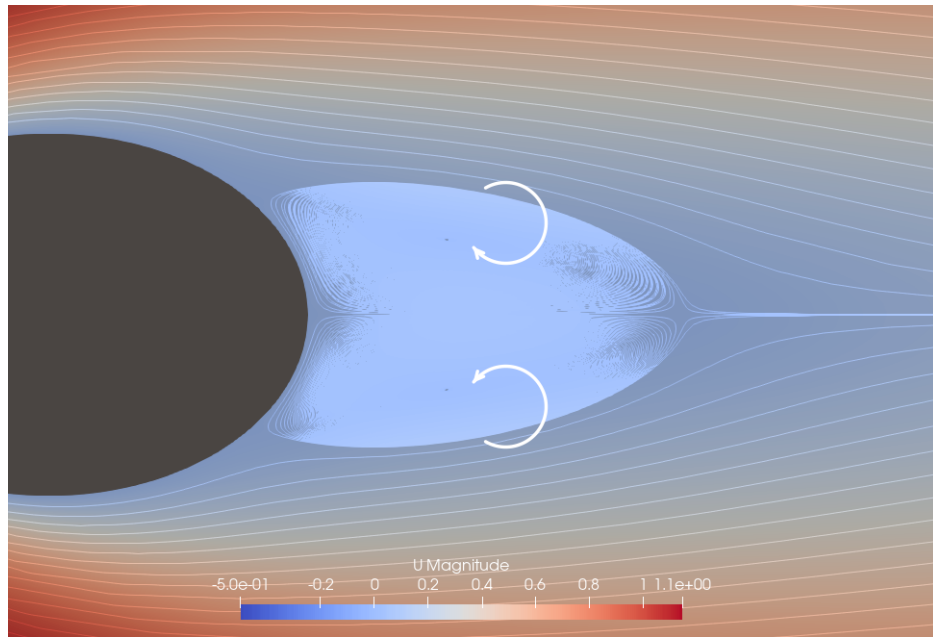


Figura 5.6: Bolla di ricircolazione nella scia del cilindro ($Re = 40$), visualizzata tramite il criterio LIC (Line Integral Convolution). Le frecce bianche indicano il verso di rotazione dei due vortici simmetrici controrotanti. La bolla si estende sino a $x \approx 3R$ a valle della superficie, definendo la zona in cui il calcolo dell'errore relativo risulta instabile.

- `line_search_method`: "filter": ricerca in linea con strategia a filtro, più robusta della penalizzazione classica per problemi non convessi.
- `alpha_red_factor`: 0.5: fattore di riduzione del passo nella backtracking line search.

Capitolo 6

Analisi Parametrica e Studio di Fattibilità

Questo capitolo costituisce il cuore metodologico della tesi. Dopo aver stabilito nei capitoli precedenti l'impianto teorico (le equazioni di governo e la funzione obiettivo, Capitolo 2), il metodo dell'aggiunto discreto (Capitolo 3) e la sua implementazione pratica (Capitoli 4–5), si affronta ora la domanda centrale del lavoro: *quali condizioni di misura rendono possibile (e quanto accurata) la ricostruzione inversa della forma di un corpo tozzo a partire dalla sua firma di scia?* La risposta non è unica, perché il problema è mal posto: dipende dalla posizione dei sensori rispetto alle strutture della scia, dal loro numero, dal numero di Reynolds, dalla formulazione dell'obiettivo e, in ultima analisi, dalla quantità di informazione geometrica che il campo di velocità trasporta nella regione campionata.

Per rispondere in modo quantitativo si è condotta un'ampia campagna di ottimizzazioni inverse sul caso di riferimento (deformazione di un'ellisse $a = 1$, $b = 0.7$ verso un cilindro di raggio $R = 1$ in flusso laminare), variando sistematicamente un parametro alla volta. Complessivamente sono stati processati diciotto casi validi, che coprono: sei configurazioni di posizione e numero di probes nelle due finestre principali di scia (Sezione 6.3); il limite inferiore rappresentato dalla scia prossima e dalla bolla di ricircolazione (Sezione 6.4); la sensibilità al numero di Reynolds (Sezione 6.5); il confronto tra due formulazioni della funzione obiettivo (Sezione 6.6); e infine una serie di casi estremi che ne saggiano i confini di applicabilità (Sezione 6.7). Ogni configurazione è valutata non solo tramite la convergenza della funzione obiettivo, ma soprattutto attraverso una metrica geometrica indipendente, l'errore quadratico medio radiale rispetto al cilindro target, introdotta nella Sezione 6.2. La distinzione tra questi due indicatori, cioè la discrepanza dei campi ai probes e la fedeltà geometrica della forma ricostruita, è uno dei fili conduttori dell'intero capitolo.

Il capitolo ha dunque una duplice natura. Da un lato è uno *studio di fattibilità*, che stabilisce se e in quali condizioni l'inverse design basato sulla scia converga a una geometria corretta; dall'altro è un'analisi di sensibilità, che quantifica come la qualità della ricostruzione risponda alle scelte di progetto. Le conclusioni operative raccolte nella sintesi finale (Sezione 6.8) motivano la configurazione adottata come caso base nel Capitolo 7, dove la stessa ottimizzazione viene analizzata in profondità dal punto di vista fluidodinamico.

6.1 Indipendenza dalla griglia computazionale

Nel presente lavoro la mesh computazionale non è stata oggetto di uno studio sistematico di indipendenza dalla griglia. Tale scelta, apparentemente in controtendenza rispetto alle buone pratiche della CFD, è invece pienamente giustificata dalla natura specifica del problema affrontato ed è supportata da due considerazioni principali.

La prima è di natura metodologica: il problema affrontato è un *inverse crime* consapevole, in cui sia il campo di velocità di riferimento $\mathbf{U}^{\text{ref}}$ (ottenuto dalla simulazione sul cilindro target) sia tutti i campi calcolati durante il ciclo di ottimizzazione utilizzano la stessa identica discretizzazione. Qualsiasi errore sistematico introdotto dalla griglia è quindi identico in $\mathbf{U}$ e in $\mathbf{U}^{\text{ref}}$, e si cancella nella definizione dell'errore relativo $\varepsilon_{ik} = (U_k - U_k^{\text{ref}})/U_k^{\text{ref}}$. Di conseguenza, la funzione obiettivo e i gradienti aggiuntivi non risentono

dell'errore di discretizzazione assoluto, ma solo delle differenze geometriche tra i due corpi.

La seconda motivazione è di natura fisica: a $Re = 40$ il flusso è laminare, stazionario e privo di strutture turbolente. In questo regime la mesh O-strutturata $100 \times 100 \times 1$ celle per blocco (40 000 celle totali), con raffinamento parietale $1/500$, è ampiamente sovradimensionata rispetto ai requisiti minimi per la risoluzione accurata del campo di moto. I parametri di qualità della mesh [non-ortogonalità massima 12, skewness 0.13, aspect ratio 300.6 (ben al di sotto della soglia critica di 1000)] confermano l'adeguatezza della discretizzazione (si veda la Tabella 5.2). Inoltre, la convergenza primale è portata a $\|\mathbf{R}\|_\infty < 10^{-13}$, garantendo che l'errore di iterazione del solutore SIMPLE sia trascurabile rispetto a qualsiasi effetto di griglia.

È utile precisare il senso in cui l'*inverse crime* agisce da regolarizzatore dell'errore di discretizzazione. In un problema diretto convenzionale, l'obiettivo è calcolare il campo di moto "vero", e l'errore di griglia è la distanza tra la soluzione discreta e quella continua: ridurlo richiede l'infittimento della mesh fino alla convergenza asintotica. Nel problema inverso qui formulato, invece, l'unica grandezza che entra nell'ottimizzazione è la *differenza* tra due campi calcolati sulla stessa mesh (quello corrente e quello di riferimento) alle medesime posizioni dei probes. Indicando con $\mathbf{U}_h$ la soluzione discreta su griglia di passo h e con $\mathbf{U}$ la soluzione esatta, si ha $\mathbf{U}_h = \mathbf{U} + \mathbf{e}_h$, dove $\mathbf{e}_h$ è l'errore di troncamento. Poiché sia il campo corrente sia quello target condividono lo stesso operatore discreto e la stessa mesh, i rispettivi errori $\mathbf{e}_h$ sono fortemente correlati e si elidono in larga parte nella differenza che definisce l'obiettivo. La ricostruzione geometrica non converge quindi verso "la forma che genera il campo continuo vero", ma verso "la forma che, sulla stessa mesh, riproduce il campo di riferimento": un obiettivo che è per costruzione insensibile all'errore di griglia assoluto.

Questa proprietà ha una conseguenza pratica rilevante. Un eventuale studio di indipendenza dalla griglia dovrebbe essere condotto non sul singolo campo di moto, ma sull'intero ciclo di ottimizzazione, rigenerando per ciascuna densità di mesh sia il campo di riferimento sia la deformazione FFD e verificando la stabilità dell'RMSE geometrico finale. Un simile studio moltiplicherebbe il costo computazionale già considerevole della campagna parametrica senza aggiungere informazione utile, poiché per l'argomento di cancellazione appena esposto l'RMSE finale dipende dalle differenze geometriche tra i corpi e dalla risoluzione della parametrizzazione FFD (5 stazioni di controllo), non dalla densità della mesh fluida, purché quest'ultima sia sufficiente a risolvere il campo laminare. La qualità della mesh, documentata nella Tabella 5.2, garantisce ampiamente questa condizione.

Va infine osservato che il vero fattore limitante dell'accuratezza geometrica non è la mesh fluida ma la *parametrizzazione della forma*: come si vedrà nella Sezione 6.2, l'errore residuo caratteristico di tutte le ricostruzioni ha origine nel numero finito di gradi di libertà della FFD, non nella discretizzazione del dominio. Infittire la mesh fluida non ridurrebbe tale errore, mentre aumentare il numero di stazioni FFD lo farebbe, al prezzo, però, di uno spazio di progetto più ampio e di un problema inverso potenzialmente più mal condizionato. Il bilanciamento tra ricchezza della parametrizzazione e stabilità dell'ottimizzazione esula dagli obiettivi della presente analisi, che mantiene fissa la FFD a 5 stazioni per tutti i casi, isolando così l'effetto delle sole variabili di misura.

Per le ragioni esposte, la mesh descritta nel Capitolo 5 è considerata adeguata per tutti i casi analizzati nel seguito, e non si rende necessaria una variazione della densità di griglia.

6.2 Metrica di valutazione della ricostruzione geometrica

Il valore finale della funzione obiettivo f_{fin} non è, di per sé, una misura affidabile della qualità della ricostruzione geometrica. Esso quantifica infatti la discrepanza dei campi di velocità *alle posizioni dei probes*, non la distanza tra la forma ottimizzata e la geometria target. Poiché il problema è mal posto (Capitolo 2), configurazioni di misura diverse possono raggiungere valori di f molto simili pur corrispondendo a geometrie sensibilmente diverse, e viceversa. Per valutare in modo indipendente la bontà di ciascuna ricostruzione si introduce quindi una metrica puramente geometrica, calcolata a posteriori sul profilo finale del corpo.

Il profilo ottimizzato viene estratto dalla mesh deformata dell'ultima iterazione tramite lo script `geometric_error.py`, che legge le facce della superficie del corpo dai file `constant/polyMesh/faces` e le coordinate dei nodi dal corrispondente file `points` dell'ultimo istante temporale. I punti della superficie vengono espressi in coordinate polari (r_j, θ_j) centrate sull'asse del corpo, ottenendo il raggio ottimizzato $r_{\text{opt}}(\theta)$ come funzione dell'angolo. Poiché la geometria target è un cilindro di raggio unitario, il raggio di riferimento è costante, $r_{\text{target}}(\theta) = R = 1$, e l'errore radiale locale è semplicemente $e(\theta) = r_{\text{opt}}(\theta) - 1$.

La qualità della ricostruzione è sintetizzata da tre indicatori scalari:

$$\text{RMSE} = \sqrt{\frac{1}{N_\theta} \sum_{j=1}^{N_\theta} (r_{\text{opt},j} - 1)^2}, \quad \text{MAE} = \frac{1}{N_\theta} \sum_{j=1}^{N_\theta} |r_{\text{opt},j} - 1|, \quad e_{\text{max}} = \max_j |r_{\text{opt},j} - 1| \quad (6.1)$$

dove N_θ è il numero di punti della superficie. Il primo, l'errore quadratico medio radiale (RMSE), è la metrica di riferimento adottata nel seguito: essendo espresso nelle stesse unità del raggio ($R = 1$), un valore $\text{RMSE} = 0.02$ corrisponde a uno scostamento medio del 2% del raggio del cilindro. Il secondo (MAE) è meno sensibile ai picchi locali, mentre il terzo (e_{max}) individua l'errore massimo e la sua posizione angolare, utile per diagnosticare *dove* la ricostruzione è meno accurata.

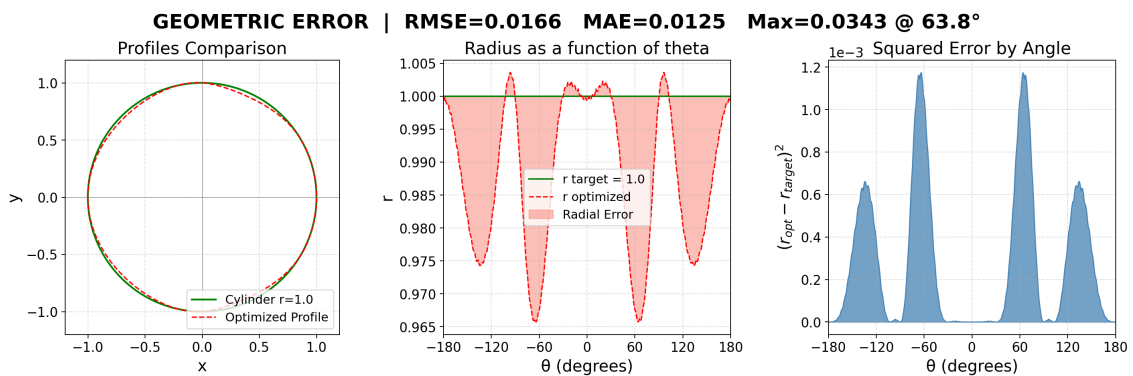


Figura 6.1: Esempio di analisi dell'errore geometrico (caso *Probes3_6*, $x \in [3R, 6R]$, 15×15 probes). Sinistra: sovrapposizione del profilo ottimizzato (rosso tratteggiato) e del cilindro target (verde). Centro: raggio $r_{\text{opt}}(\theta)$ in funzione dell'angolo, con l'area di errore radiale evidenziata. Destra: errore quadratico $(r_{\text{opt}} - r_{\text{target}})^2$ in funzione dell'angolo. I quattro lobi di errore, simmetrici rispetto agli assi, corrispondono alle zone $\theta \approx \pm 60^\circ$ e $\theta \approx \pm 120^\circ$ dove la deformazione FFD fatica a riprodurre la curvatura circolare.

La Figura 6.1 illustra il formato di questa analisi per un caso rappresentativo. Il pannello centrale mostra come l'errore radiale non sia uniforme: la deformazione FFD, parametrizzata da sole 5 stazioni longitudinali (Sezione 5.2), riproduce molto bene i punti di ristagno anteriore e posteriore ($\theta = 0, \pm 180$) e i punti di massima ampiezza trasversale ($\theta = \pm 90$), ma lascia un errore residuo caratteristico a $\theta \approx \pm 60$ e $\theta \approx \pm 120$, dove la curvatura del cilindro richiede un raggio intermedio non catturato esattamente dalle funzioni di forma. Questa firma a quattro lobi si ritrova, con ampiezze diverse, in quasi tutti i casi analizzati ed è una limitazione intrinseca della parametrizzazione geometrica, non del posizionamento dei probes.

6.2.1 Scelta della metrica e significato dei tre indicatori

La scelta di una metrica radiale $r(\theta)$ merita una giustificazione. Alternative comuni per misurare la distanza tra due curve chiuse sono la distanza di Hausdorff (massimo scostamento punto-curva), l'errore d'area (differenza tra le superfici racchiuse) o la distanza media punto-punto lungo la normale. La metrica radiale adottata qui sfrutta il fatto che il target è un cerchio centrato nell'origine: per esso $r_{\text{target}}(\theta) = R$ è costante, e la deformazione FFD conserva la simmetria e la stella-convessità del profilo, garantendo che $r_{\text{opt}}(\theta)$ sia una funzione a un solo valore dell'angolo. In queste condizioni la rappresentazione polare è priva di ambiguità e l'errore radiale $e(\theta) = r_{\text{opt}}(\theta) - R$ è direttamente interpretabile come “quanto localmente la forma è troppo gonfia o troppo magra”. Poiché tutte le quantità sono normalizzate da $R = 1$, gli indicatori sono adimensionali e confrontabili tra casi.

I tre indicatori della (6.1) forniscono informazioni complementari e vanno letti insieme. L'RMSE penalizza quadraticamente gli scostamenti e riflette l'errore “tipico” della ricostruzione, dando peso maggiore alle zone di forte deviazione; è la metrica sintetica di riferimento. Il MAE, mediando i valori assoluti, è più robusto rispetto a picchi localizzati e, confrontato con l'RMSE, indica quanto l'errore sia concentrato: un rapporto RMSE/MAE prossimo a 1 segnala un errore distribuito uniformemente, mentre un rapporto elevato tradisce la presenza di pochi lobi dominanti. L'errore massimo e_{max} , con la sua posizione angolare, localizza la peggiore deviazione e permette di distinguere ricostruzioni con lo stesso RMSE ma diversa distribuzione dell'errore; ad esempio una forma leggermente e uniformemente sbagliata da una quasi perfetta salvo una rientranza locale. Nel seguito i tre indicatori sono riportati congiuntamente in tabella, con l'RMSE come criterio principale di confronto.

6.2.2 Origine dell'errore residuo: i limiti della parametrizzazione FFD

La firma a quattro lobi visibile in Figura 6.1 non è un artefatto ma la manifestazione geometrica dei gradi di libertà finiti della FFD. La deformazione è controllata da cinque stazioni longitudinali equispaziate in $x \in \{-1, -0.5, 0, 0.5, 1\}$, ciascuna delle quali sposta simmetricamente la coppia di punti di controllo superiore e inferiore. Il profilo risultante è quindi una curva la cui ordinata è, in ogni sezione x , interpolata tra i cinque valori di controllo tramite le funzioni di base B-spline del box FFD. Riprodurre esattamente un cerchio richiederebbe un raggio che varia con continuità secondo $y = \sqrt{R^2 - x^2}$ nella metà superiore; le cinque stazioni catturano fedelmente i punti vincolati: i due estremi ($x = \pm 1$, dove $y = 0$) e il centro ($x = 0$, dove $y = R$), ma nelle stazioni intermedie ($x = \pm 0.5$)

l'interpolazione B-spline non coincide perfettamente con l'arco circolare, e la differenza si proietta sugli angoli $\theta \approx \pm 60$ e $\theta \approx \pm 120$, dove si osservano i lobi di errore.

Questa interpretazione è confermata da due fatti ricorrenti in tutta la campagna: primo, la posizione angolare dei lobi è pressoché invariante rispetto al posizionamento dei probes, poiché dipende dalla geometria della parametrizzazione e non dalla misura; secondo, il segno dell'errore nei lobi può cambiare (raggio localmente in eccesso o in difetto) a seconda di come l'ottimizzatore distribuisce la deformazione tra le stazioni, ma la loro collocazione resta stabile. Ne consegue che esiste un *errore di ricostruzione minimo strutturale*, non abbattibile agendo sui probes, il cui ordine di grandezza è fissato dalla risoluzione della FFD. I valori di RMSE più bassi ottenuti nella campagna (~ 0.012) sono verosimilmente prossimi a questo limite, il che spiega perché, oltre una certa qualità della misura, ulteriori miglioramenti della configurazione dei probes producano guadagni marginali. Questa osservazione sarà ripresa più volte nel seguito e costituisce la chiave interpretativa di molti dei risultati parametrici.

6.3 Studio parametrico del posizionamento dei probes

L'analisi parametrica è organizzata attorno a due dimensioni: la **posizione** della finestra di misura (scia intermedia $x \in [3R, 6R]$ contro scia lontana $x \in [6R, 9R]$) e il **numero di probes** per finestra, ottenuto restringendo progressivamente l'estensione longitudinale della griglia. Partendo dalla griglia piena $15 \times 15 = 225$ probes che copre una finestra di $3R \times 3R$, si ricavano due varianti ridotte mantenendo le prime 8 stazioni longitudinali ($8 \times 15 = 120$ probes) e le prime 4 ($4 \times 15 = 60$ probes); il passo trasversale e il range in $y \in [-1.5R, 1.5R]$ restano invariati. Per ciascuna delle sei combinazioni si valutano la convergenza della funzione obiettivo f e l'errore geometrico radiale rispetto al cilindro target.

I casi considerati, tutti con ellisse di partenza $a = 1$, $b = 0.7$, ottimizzatore IPOPT e funzione obiettivo *VarianceScaled* a due componenti ($k \in \{0, 1\}$), sono riassunti nella Tabella 6.1. La Figura 6.2 mostra il posizionamento delle due griglie piene nella scia.

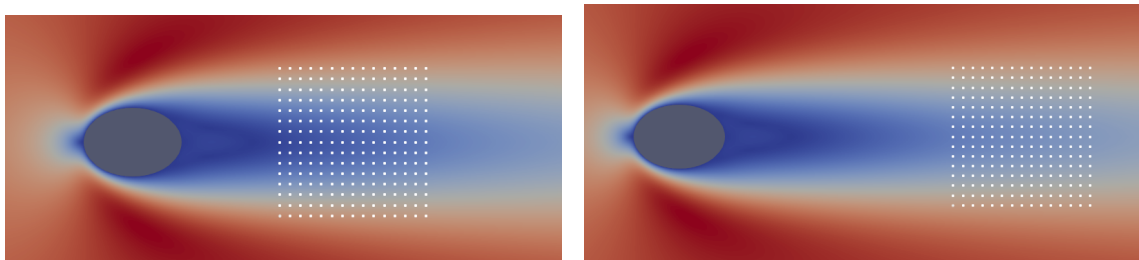


Figura 6.2: Disposizione dei probes (punti) nella scia dell'ellisse per le due finestre principali dello studio parametrico, griglia piena 15×15 : scia intermedia $x \in [3R, 6R]$ (sinistra) e scia lontana $x \in [6R, 9R]$ (destra). Sullo sfondo il campo di velocità di riferimento del cilindro target.

6.3.1 Panoramica dei sei casi

Prima di analizzare le tendenze, conviene descrivere brevemente il comportamento di ciascuna delle sei configurazioni, la cui sintesi numerica è nella Tabella 6.1.

Tabella 6.1: Riepilogo dei casi dell'analisi parametrica posizione/numero. N : numero di probes; $Iter$: numero di iterazioni IPOPT alla convergenza; f_{fin} : valore finale dell'obiettivo; RMSE, MAE, e_{max} : metriche geometriche definite in (6.1), in unità di $R = 1$. Tutti i casi: ellisse $b = 0.7$, *VarianceScaled* a due componenti, $Re = 40$.

Caso	N	$x_{min}-x_{max}$	Griglia	Iter	f_{fin}	RMSE	MAE	e_{max}
<i>Scia intermedia</i>								
Probes3_37	60	$[3.0, 3.7]R$	4×15	44	1.3×10^{-3}	0.0384	0.0319	0.066
Probes3_45	120	$[3.0, 4.5]R$	8×15	52	3.7×10^{-3}	0.0279	0.0227	0.049
Probes3_6	225	$[3.0, 6.0]R$	15×15	65	8.8×10^{-3}	0.0166	0.0125	0.034
<i>Scia lontana</i>								
Probes6_67	60	$[6.0, 6.7]R$	4×15	53	1.3×10^{-4}	0.0224	0.0168	0.047
Probes6_75	120	$[6.0, 7.5]R$	8×15	44	3.1×10^{-5}	0.0130	0.0096	0.028
Probes6_9	225	$[6.0, 9.0]R$	15×15	56	1.5×10^{-4}	0.0054	0.0041	0.012

Nella *scia intermedia* ($x \in [3R, 6R]$), il caso più povero è Probes3_37 (60 probes, finestra $[3, 3.7]R$): l'ottimizzazione converge in 44 iterazioni a $f_{fin} = 1.3 \times 10^{-3}$, ma la ricostruzione geometrica si ferma a RMSE = 0.0384, con l'errore massimo (0.066) localizzato a $\theta \approx 143$, nella regione posteriore. Ampliando la finestra a $[3, 4.5]R$ (Probes3_45, 120 probes) l'RMSE scende a 0.0279 e il picco d'errore si sposta ai fianchi ($\theta \approx 62$). Con la griglia piena $[3, 6]R$ (Probes3_6, 225 probes) si raggiunge il miglior risultato della zona intermedia, RMSE = 0.0166, con la firma a quattro lobi ben bilanciata e un errore massimo contenuto a 0.034.

Nella *scia lontana* ($x \in [6R, 9R]$) l'andamento è analogo ma i valori sono complessivamente migliori. Il caso a 60 probes Probes6_67 ($[6, 6.7]R$) ottiene RMSE = 0.0224, già migliore del corrispondente intermedio a 60 probes; a 120 probes (Probes6_75, $[6, 7.5]R$) l'RMSE scende a 0.0130 con un errore massimo di soli 0.028 ben distribuito; e la griglia piena Probes6_9 ($[6, 9]R$, 225 probes) raggiunge il miglior risultato tra tutti i casi principali, RMSE = 0.0054 cioè appena mezzo punto percentuale del raggio, convergendo peraltro in sole 56 iterazioni. È notevole che quest'ultimo caso, pur avendo l'RMSE più basso, *non* presenti il valore di f_{fin} più basso della sua famiglia: quel primato spetta a Probes6_75 ($f_{fin} = 3.1 \times 10^{-5}$ contro 1.5×10^{-4} di Probes6_9), che tuttavia ricostruisce una geometria meno accurata. Questa apparente contraddizione introduce il tema centrale della sezione seguente.

6.3.2 Disaccoppiamento tra obiettivo e accuratezza geometrica

Il risultato più importante che emerge dalla Tabella 6.1 è che il valore finale della funzione obiettivo e l'accuratezza geometrica misurata dall'RMSE non sono legati da una relazione semplice: un f_{fin} più basso non implica una ricostruzione migliore. Due osservazioni lo rendono evidente. La prima riguarda la serie in scia intermedia: allargando la finestra da $[3, 3.7]R$ a $[3, 6]R$ il valore finale dell'obiettivo *aumenta* (da 1.3×10^{-3} a 8.8×10^{-3}), mentre l'RMSE *migliora* (da 0.0384 a 0.0166). Obiettivo e geometria si muovono qui in direzioni opposte. La seconda osservazione è un confronto incrociato: il caso in scia lontana a 60 probes (Probes6_67) raggiunge $f_{fin} = 1.3 \times 10^{-4}$, un valore circa settanta volte inferiore a quello del miglior caso intermedio (Probes3_6, $f_{fin} = 8.8 \times 10^{-3}$), eppure ricostruisce una geometria *peggiore* (RMSE = 0.0224 contro 0.0166). Più in generale,

attraverso i sei casi il valore finale dell'obiettivo varia di oltre due ordini di grandezza (da $\sim 10^{-2}$ in scia intermedia a $\sim 10^{-5}$ in scia lontana), mentre l'RMSE resta confinato entro un fattore sette (0.005–0.038): le due grandezze hanno sensibilità e dinamiche del tutto diverse.

La ragione di questo comportamento è fisica. Nella scia lontana il campo di velocità è più uniforme e regolare: la firma geometrica del corpo è già in gran parte diffusa, quindi campi generati da forme diverse differiscono poco ai probes, e l'ottimizzatore raggiunge facilmente valori di f molto piccoli. Ma la stessa debolezza della sensibilità rende l'ultimo tratto di avvicinamento alla geometria esatta lento e faticoso; è per questo che il caso ottimale Probes6_9 impiega 56 iterazioni, gran parte delle quali in un lento affinamento finale e fa sì che, spingendosi troppo a valle, l'informazione geometrica non basti più a rifinire la forma: la ricostruzione degrada (come si vedrà per i casi a 21–27R) anche mentre f continua a diminuire. Nella scia intermedia accade il contrario: il campo è ancora fortemente modellato dalla geometria, per cui f resta più alto (le differenze residue ai probes sono amplificate), ma i gradienti sono più ricchi e guidano la forma con maggiore efficacia per ogni unità di obiettivo ridotta. In sintesi, *un valore basso della funzione obiettivo non garantisce di per sé una buona ricostruzione geometrica*: è questa la ragione per cui l'intera analisi parametrica si fonda sull'RMSE e non su f_{fin} .

Il fenomeno può essere inquadrato formalmente nei termini del condizionamento del problema inverso. La funzione obiettivo mappa lo spazio delle forme Ω nello spazio degli scarti ai probes; la sua sensibilità alla geometria è misurata, in prima approssimazione, dalla norma del gradiente $\partial f / \partial \mathbf{x}$ e più in generale dalla curvatura della funzione costo attorno al minimo. Nella scia lontana questa curvatura è bassa: il minimo è ampio e piatto, per cui variazioni geometriche non trascurabili spostano di poco il valore di f . L'ottimizzatore raggiunge facilmente un f molto piccolo, ma l'ultimo tratto di avvicinamento alla geometria esatta avviene lungo direzioni a cui l'obiettivo è quasi insensibile: sono, in sostanza, direzioni prossime al nucleo dell'operatore, lungo le quali il problema è mal condizionato e il gradiente porta poca informazione. La ricostruzione si arresta quindi su un plateau geometrico determinato non dalla precisione con cui si annulla f , ma dal contenuto informativo del campo. Nella scia intermedia la curvatura è maggiore, il minimo più stretto e i gradienti più informativi: f_{fin} resta più alto perché il campo è più sensibile alla forma e le differenze residue ai probes sono amplificate, ma proprio questa sensibilità consente una ricostruzione competitiva. La distinzione tra “ridurre l'obiettivo” e “ricostruire la geometria” è dunque una manifestazione diretta della mal-posizione del problema, e non un artefatto numerico.

Una conseguenza pratica di questa analisi è che il valore di f_{fin} non dovrebbe mai essere usato, isolatamente, come criterio di confronto tra configurazioni di misura diverse né, a maggior ragione, come criterio di arresto “fisico” dell'ottimizzazione. Nel presente lavoro l'arresto è governato dalle tolleranze KKT interne di IPOPT (Sezione 5.3), mentre la valutazione comparativa della qualità è affidata all'RMSE geometrico calcolato a posteriori. In un contesto applicativo, in cui la geometria vera non è nota e l'RMSE non è calcolabile, questa osservazione suggerisce cautela: due misure che producono valori di obiettivo simili possono corrispondere a ricostruzioni di qualità molto diversa, e la sola convergenza di f non è garanzia di successo.

Il fenomeno è reso evidente dal confronto diretto di tutte le posizioni esplorate, dalla scia prossima ($x \in [1.5R, 3R]$) fino all'estremo del dominio ($x \in [24R, 27R]$), riportato

in Figura 6.3. Le due curve hanno andamenti radicalmente diversi. Il valore finale dell'obiettivo (asse destro, logaritmico) *decreisce* in modo pressoché monotono con la distanza dei probes, di oltre quattro ordini di grandezza, da $f_{\text{fin}} \approx 0.5$ nella scia prossima, a 1.5×10^{-4} in [6, 9], fino a $\sim 5 \times 10^{-6}$ nelle finestre lontanissime, riflettendo il progressivo appiattimento della funzione costo man mano che il campo diventa più uniforme e facile da riprodurre ai probes. L'RMSE geometrico (asse sinistro, lineare), invece, resta confinato entro un fattore diciotto e segue una curva a U: elevatissimo nella scia prossima (0.0906, per l'instabilità discussa nella Sezione 6.4), scende al minimo di [6, 9] (0.0054), poi risale verso le finestre lontanissime (0.0248–0.0252). I due indicatori sono quindi *anticorrelati nella scia lontana*: oltre l'ottimo di [6, 9], f_{fin} continua a scendere (di circa trenta volte fino a [21, 24]) mentre l'RMSE peggiora di un fattore cinque. In quella regione l'obiettivo è di fatto cieco al degrado geometrico, continua a segnalare valori sempre più piccoli mentre la forma ricostruita si allontana dal cilindro. Nessuna analisi affidata al solo valore dell'obiettivo coglierebbe questo degrado; è l'esistenza di una metrica geometrica indipendente a rivelarlo.

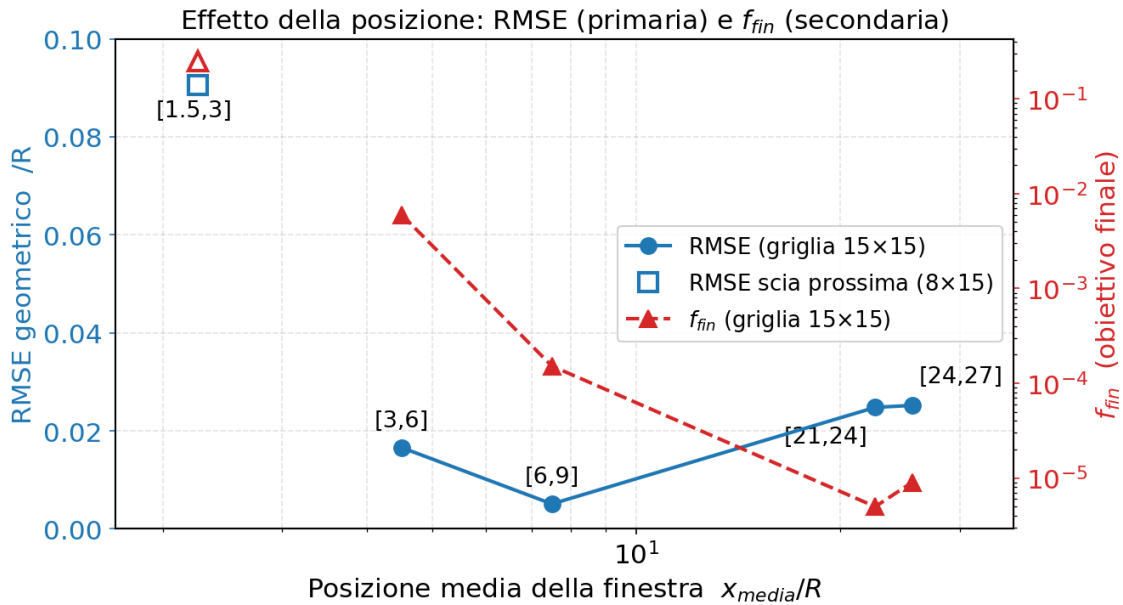


Figura 6.3: Effetto della posizione della finestra di misura sulla ricostruzione, dall'intera gamma esplorata ($x \in [1.5R, 3R]$ fino a $x \in [24R, 27R]$). Asse sinistro (blu): RMSE geometrico, metrica primaria di qualità. Asse destro (rosso, logaritmico): valore finale della funzione obiettivo f_{fin} , metrica secondaria. Mentre f_{fin} *decreisce* in modo pressoché monotono con la distanza (oltre quattro ordini di grandezza), l'RMSE segue una curva a U con minimo in [6R, 9R]: nella scia lontana i due indicatori sono anticorrelati, cioè l'obiettivo continua a scendere mentre la geometria peggiora, a riprova del fatto che l'obiettivo è “cieco” al degrado geometrico lontano. Il quadrato vuoto indica la scia prossima (griglia 8×15), gli altri punti le griglie piene 15×15 .

Lo stesso messaggio emerge dalle storie di convergenza sovrapposte (Figura 6.4), che confrontano l'evoluzione della funzione obiettivo per quattro posizioni rappresentative. Al crescere della distanza dei probes il plateau finale dell'obiettivo si abbassa sistematicamente (partendo vicino da un $f \approx 0.5$) fino alle finestre lontanissime, che raggiungono

valori dell'ordine di 10^{-6} . Il caso ottimale $[6, 9]$ converge in 56 iterazioni, il minor numero tra le quattro posizioni, ed è al tempo stesso il più accurato. Ma bisogna ricordarsi che un plateau basso non implica strettamente una buona geometria: la finestra lontanissima $[21, 24]$ raggiunge un plateau ($\sim 5 \times 10^{-6}$) inferiore a quello di $[6, 9]$ (1.5×10^{-4}), eppure la sua ricostruzione ($\text{RMSE} = 0.0248$) è cinque volte meno accurata (0.0054). Osservando le sole storie di convergenza si giudicherebbe la finestra lontanissima migliore, mentre è vero il contrario: la loro differenza vive interamente nella metrica geometrica. Le due figure, lette congiuntamente, riassumono il messaggio centrale del capitolo e giustificano l'adozione dell'RMSE come metrica di riferimento.

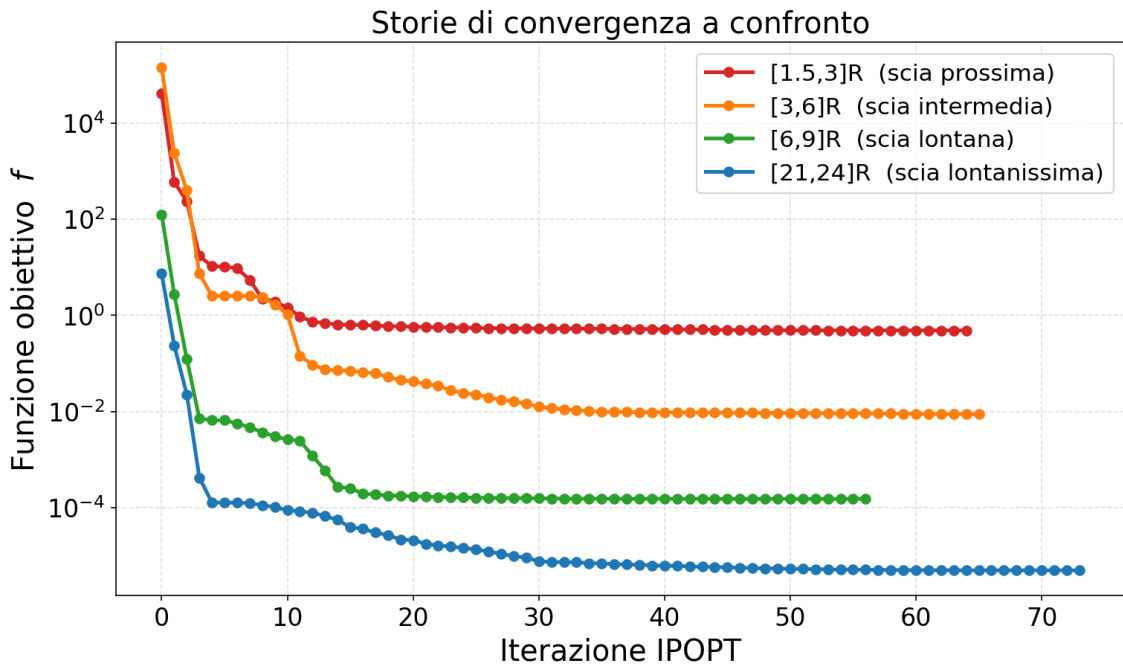


Figura 6.4: Storie di convergenza della funzione obiettivo (scala logaritmica) per quattro posizioni rappresentative della finestra di misura. Al crescere della distanza dei probes il plateau di convergenza si abbassa; la scia prossima ($x \in [1.5R, 3R]$, in rosso) resta bloccata su un plateau elevato senza convergere. Il caso ottimale $[6, 9]R$ converge nel minor numero di iterazioni (56) ed è il più accurato; tuttavia un plateau basso non implica una buona geometria: la finestra $[21, 24]R$ raggiunge un plateau perfino inferiore a quello di $[6, 9]R$ ma ricostruisce cinque volte peggio (cfr. Figura 6.3).

6.3.3 Effetto del numero di probes

A parità di finestra, l'RMSE diminuisce in modo *monotono* all'aumentare del numero di probes, ovvero all'aumentare dell'estensione longitudinale della griglia. In scia intermedia si passa da $\text{RMSE} = 0.0384$ con 60 probes ($x \in [3, 3.7]R$) a 0.0279 con 120 probes ($x \in [3, 4.5]R$) fino a 0.0166 con 225 probes ($x \in [3, 6]R$); in scia lontana da 0.0224 (60 probes) a 0.0130 (120) fino a 0.0054 (225). L'andamento è coerente con l'intuizione: una finestra più estesa campiona una porzione più ampia della scia e vincola in modo più completo la ricostruzione, riducendo l'ambiguità del problema inverso. La Figura 6.5

mostra le tre griglie della scia lontana, dal sottoinsieme più compatto (4×15) alla griglia piena (15×15).

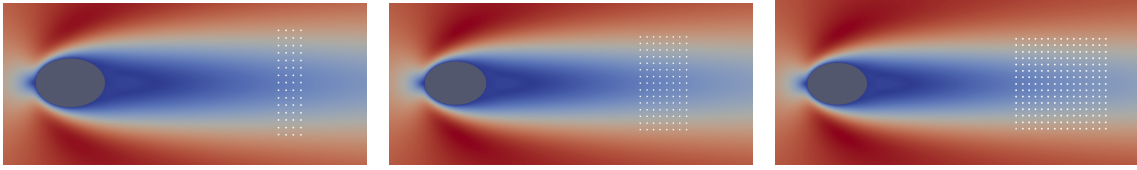


Figura 6.5: Progressiva estensione della griglia di probes in scia lontana, a parità di passo trasversale: $4 \times 15 = 60$ probes su $[6, 6.7]R$ (sinistra), $8 \times 15 = 120$ su $[6, 7.5]R$ (centro), $15 \times 15 = 225$ su $[6, 9]R$ (destra). L'RMSE geometrico migliora monotonamente al crescere della finestra ($0.0224 \rightarrow 0.0130 \rightarrow 0.0054$), con il salto più marcato proprio nel passaggio dalla griglia a 120 a quella piena a 225 probes.

Va osservato che, contrariamente a quanto si potrebbe supporre, il guadagno non si esaurisce con l'aumento dei sensori: in scia lontana il salto più consistente si registra proprio all'ultimo passo, dove il passaggio da 120 a 225 probes migliora l'RMSE del 60% ($0.0130 \rightarrow 0.0054$). La griglia piena estrae dunque informazione geometrica ancora significativa dalle stazioni più a valle, e la finestra estesa fino a $9R$ è pienamente utile: il punto di rendimenti marginali non è stato raggiunto entro i 225 probes esplorati. La configurazione a 120 probes resta un buon compromesso quando il costo computazionale è un vincolo, ma lascia sul tavolo una parte apprezzabile di accuratezza. All'estremo opposto, comprimere la finestra a soli 60 probes degrada sensibilmente la ricostruzione (RMSE più che quadruplicato in scia lontana, $0.0054 \rightarrow 0.0224$), segnalando che quattro sole stazioni longitudinali non catturano informazione sufficiente.

Il confronto diretto delle due famiglie è riassunto in Figura 6.6, che riporta l'RMSE in funzione del numero di probes per le finestre in scia intermedia e lontana. Emergono con chiarezza tre fatti: la scia lontana è sistematicamente più accurata della intermedia a parità di griglia (la curva verde giace sotto quella arancione); entrambe migliorano monotonamente con il numero di sensori; ed entrambe continuano a migliorare fino alla griglia piena, senza saturare entro i 225 probes. La scia lontana, in particolare, mostra la discesa più ripida nell'ultimo tratto, raggiungendo con la griglia 15×15 il miglior risultato assoluto dello studio (0.0054). Questo comportamento riflette il fatto che, in entrambe le regioni, ogni stazione longitudinale aggiuntiva porta informazione geometrica ancora sfruttabile; l'estensione longitudinale della misura si conferma il parametro più efficace per migliorare la ricostruzione, più del semplice infittimento a parità di finestra.

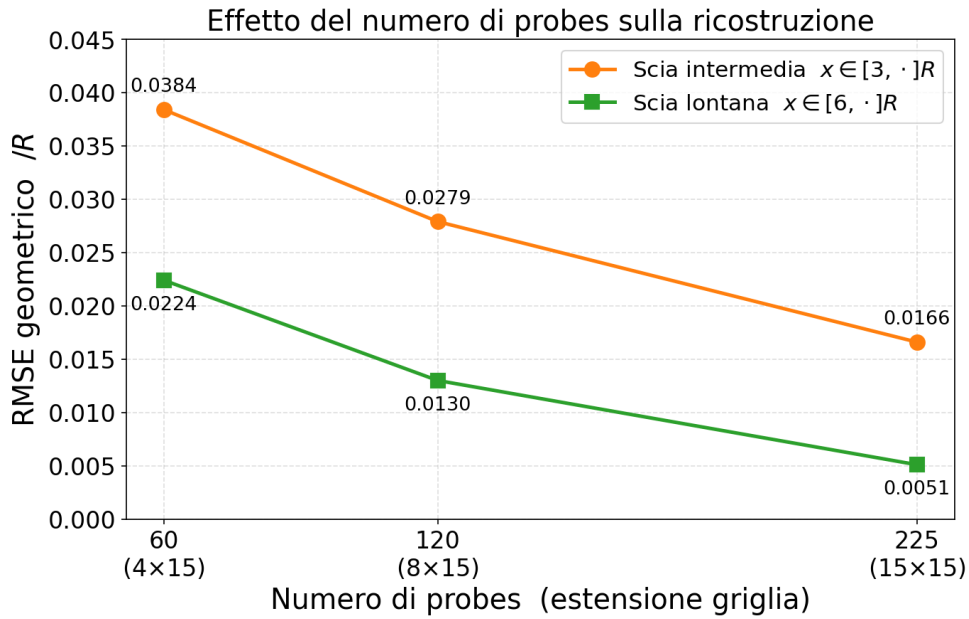


Figura 6.6: RMSE geometrico in funzione del numero di probes (estensione longitudinale della griglia, da $4 \times 15 = 60$ a $15 \times 15 = 225$) per le due finestre principali. La scia lontana ($x \in [6, \cdot]R$, verde) è sistematicamente più accurata della intermedia ($x \in [3, \cdot]R$, arancio); entrambe migliorano monotonamente fino alla griglia piena, e la scia lontana mostra la discesa più ripida nell'ultimo tratto raggiungendo il miglior risultato dello studio (0.0054).

L'effetto dell'estensione della griglia sulla serie in scia lontana è documentato in dettaglio nella Figura 6.7, che affianca a ciascuna ricostruzione geometrica (sinistra) la corrispondente storia di convergenza (destra). Passando dalla griglia 4×15 alla 15×15 il profilo ottimizzato aderisce sempre meglio al cilindro target, mentre la storia di convergenza mostra plateau finali via via più bassi e, per la griglia piena, la discesa più netta.

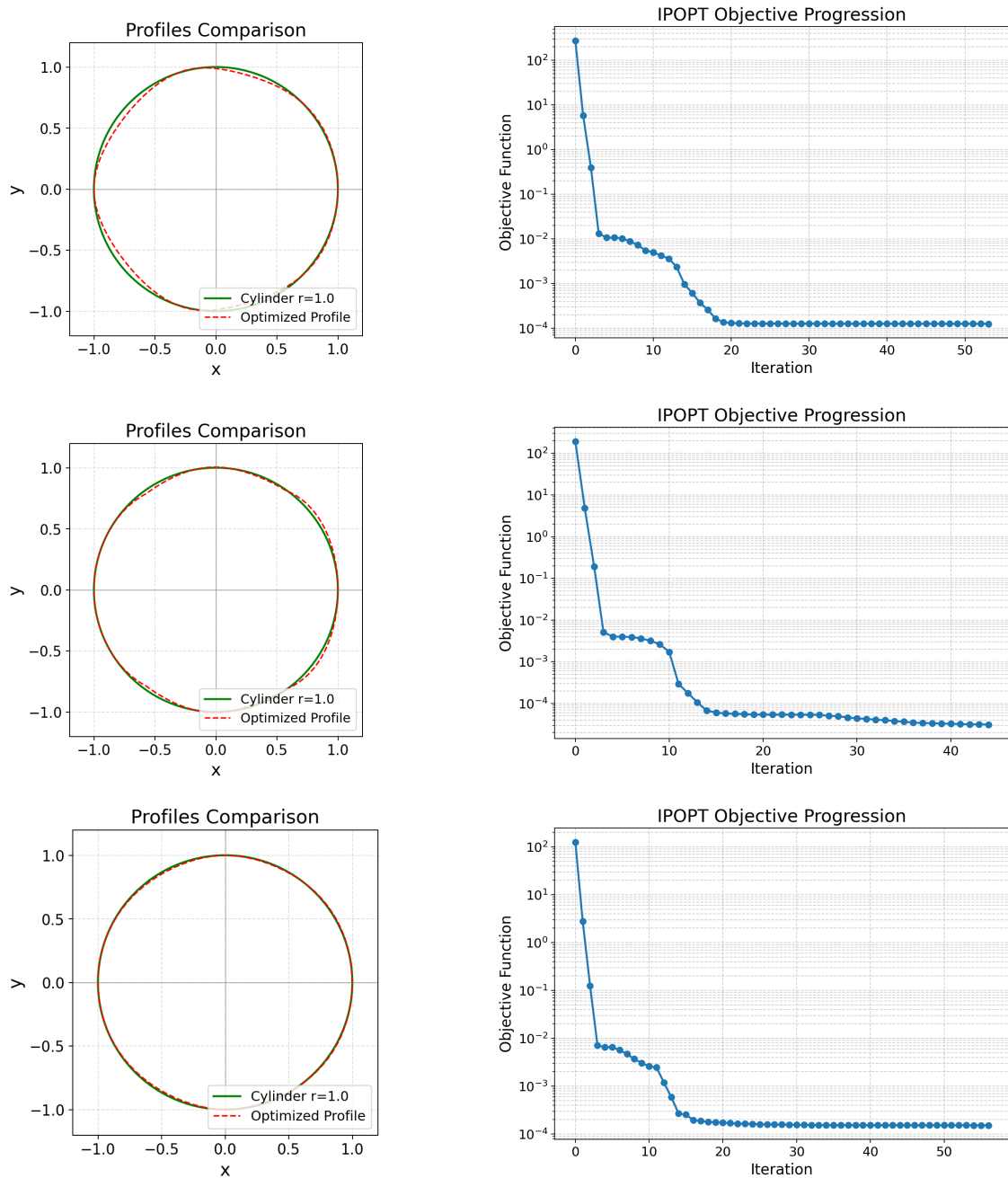


Figura 6.7: Serie in scia lontana al variare del numero di probes: ricostruzione geometrica (sinistra, profilo ottimizzato in rosso sul cilindro target in verde) e storia di convergenza IPOPT (destra). Dall'alto in basso: Probes6_67 ($4 \times 15 = 60$ probes, RMSE = 0.0224), Probes6_75 ($8 \times 15 = 120$, 0.0130), Probes6_9 ($15 \times 15 = 225$, 0.0054). All'aumentare della griglia migliorano sia l'aderenza geometrica sia il valore finale dell'obiettivo.

Un andamento del tutto analogo si osserva nella serie in scia intermedia (Figura 6.8). Anche qui l'aderenza al cilindro migliora monotonamente con l'estensione della griglia, benché i valori di RMSE restino a ogni passo superiori a quelli della corrispondente configurazione in scia lontana, coerentemente con quanto già evidenziato dalla Figura 6.6. Si noti, per contro, che le storie di convergenza della serie intermedia si arrestano su plateau

più alti ($\sim 10^{-3}$ – 10^{-2}) rispetto a quelli della serie lontana ($\sim 10^{-4}$ – 10^{-5}): un'ulteriore illustrazione del disaccoppiamento tra valore finale dell'obiettivo e accuratezza geometrica, poiché come mostra il confronto tra le due figure plateau più alti in scia intermedia si accompagnano comunque a ricostruzioni di qualità paragonabile.

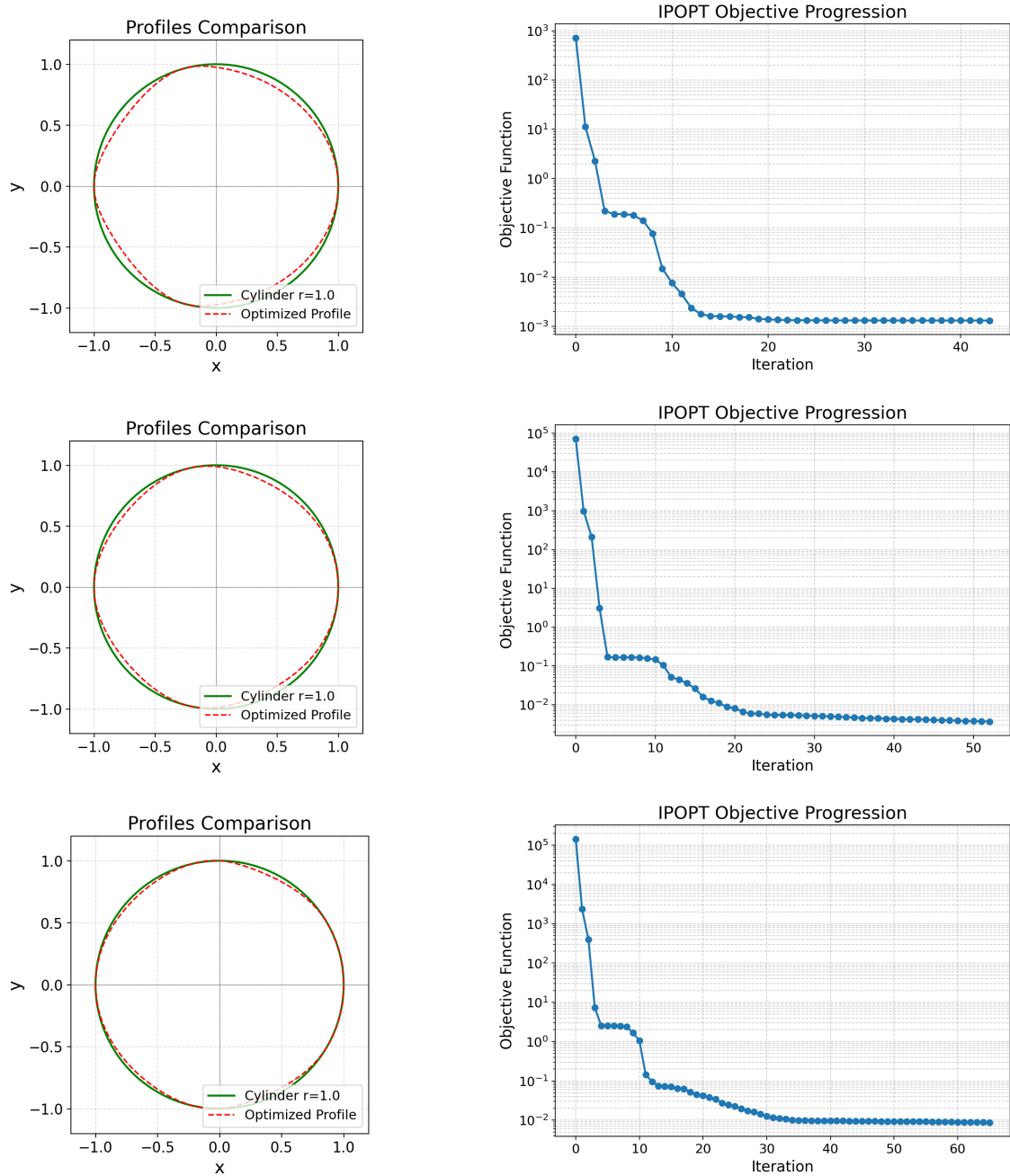


Figura 6.8: Serie in scia intermedia al variare del numero di probes: ricostruzione geometrica (sinistra, profilo ottimizzato in rosso sul cilindro target in verde) e storia di convergenza IPOPT (destra). Dall'alto in basso: Probes3_37 ($4 \times 15 = 60$ probes, RMSE = 0.0384), Probes3_45 ($8 \times 15 = 120$, 0.0279), Probes3_6 ($15 \times 15 = 225$, 0.0166). L'aderenza geometrica migliora con la griglia, mentre i plateau di convergenza restano più alti che in scia lontana.

6.3.4 Effetto della posizione: scia intermedia contro scia lontana

Confrontando le due finestre a parità di griglia, la scia lontana produce ricostruzioni geometriche nettamente migliori: per la griglia piena, RMSE = 0.0054 (lontana) contro 0.0166 (intermedia), un fattore tre a favore della scia lontana; per la griglia a 120 probes, 0.0130 contro 0.0279. Le due storie di convergenza (Figura 6.9) hanno forma simile: entrambe partono da un valore elevato dell'obiettivo e attraversano un plateau intermedio prima di stabilizzarsi, ma il caso lontano Probes6_9 ($f_0 \approx 126$) converge *più rapidamente* (56 iterazioni contro 65) e a un valore finale più basso (1.5×10^{-4} contro 8.8×10^{-3}), oltre che a una geometria più accurata. Il caso intermedio Probes3_6 parte molto più alto ($f_0 \approx 1.5 \times 10^5$) e attraversa un lungo plateau attorno a $f \approx 2 \times 10^0$ prima di scendere. I gradienti in scia intermedia sono più informativi ma più difficili da seguire per l'ottimizzatore, quelli in scia lontana più deboli ma più regolari, il che spiega la convergenza al tempo stesso più rapida e più accurata del caso lontano.

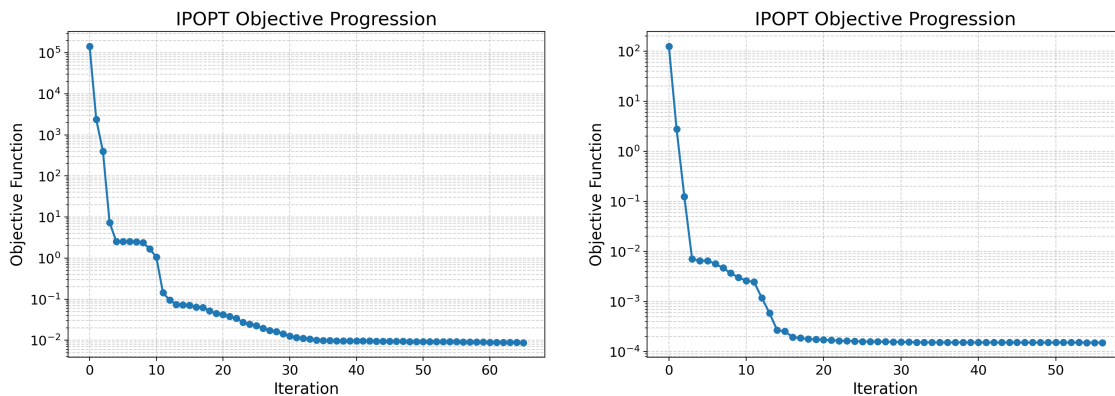


Figura 6.9: Storia della funzione obiettivo IPOPT (scala logaritmica) per i due casi a griglia piena: scia intermedia Probes3_6 (sinistra, 65 iterazioni) e scia lontana Probes6_9 (destra, 56 iterazioni). Il caso lontano converge più rapidamente e a un valore finale più piccolo; la sua ricostruzione geometrica è anche più accurata di un fattore tre (RMSE = 0.0054 contro 0.0166, cfr. Tabella 6.1).

Emerge inoltre una differenza qualitativa nella distribuzione angolare dell'errore. Nei casi in scia lontana il massimo errore tende a collocarsi nella regione posteriore-laterale ($\theta \approx \pm 120$, verso il punto di ristagno di valle), mentre nei casi in scia intermedia si sposta verso $\theta \approx \pm 60$ (fianchi del corpo). Questo riflette il fatto che probes più vicini al corpo sono più sensibili alla geometria dei fianchi, che governano l'accelerazione laterale del flusso, mentre probes lontani vincolano soprattutto la parte posteriore, che determina la struttura della scia. Nessuna delle due posizioni domina l'altra su tutto il profilo: la scia lontana vince di poco in media, ma la scia intermedia ricostruisce meglio la regione posteriore.

Questa complementarità ha una lettura fisica precisa in termini di quale porzione della superficie ciascuna zona di misura “interroga”. Il campo di velocità in un punto della scia è il risultato integrale dell'intera storia del flusso attorno al corpo, ma la sensibilità di quel punto alle diverse porzioni della superficie non è uniforme. I probes collocati subito a valle, ai fianchi della scia, risentono soprattutto dei getti laterali che si staccano dalla massima ampiezza del corpo ($\theta \approx \pm 90$) e dalla curvatura dei fianchi ($\theta \approx \pm 60$): è

lì che il flusso viene accelerato e deviato, e piccole variazioni di forma in quella regione producono variazioni misurabili di velocità nella scia intermedia. I probes lontani, invece, campionano un campo in cui i getti laterali si sono già richiusi e la struttura dominante è il deficit di scia lungo l'asse, la cui ampiezza e larghezza sono controllate dalla parte posteriore del corpo ($\theta \approx \pm 140$ fino al ristagno di valle). Ne consegue che le due finestre non sono ridondanti ma *complementari*: ciascuna vincola meglio una diversa porzione del profilo. Questa osservazione suggerisce, come naturale sviluppo, l'impiego combinato di finestre multiple riguardante un tema che esula dalla presente analisi a parametro singolo ma che le tendenze osservate rendono promettente.

Un'ultima considerazione riguarda la robustezza della convergenza in funzione della posizione. Tutti e sei i casi convergono a una geometria fisicamente sensata (nessuno collassa o diverge), ma il *percorso* verso il minimo differisce sensibilmente. I casi lontani mostrano storie di convergenza monotone e rapide; i casi intermedi presentano plateau intermedi più o meno lunghi, indice di regioni della funzione costo in cui l'ottimizzatore procede lentamente prima di trovare una direzione di discesa efficace. Questo comportamento, unito ai risultati sulla scia prossima discussi nella sezione seguente, delinea un quadro coerente: allontanandosi dal corpo la funzione obiettivo diventa progressivamente più liscia e facile da ottimizzare, ma anche meno ricca di informazione geometrica. Ciò si traduce in un compromesso che raggiunge il suo equilibrio ottimale, per il caso in esame, nelle finestre a 120–225 probes attorno a $6R$ – $9R$.

6.4 Instabilità nella scia prossima

Il limite inferiore della finestra di misura è rappresentato dal caso **Probes1'5_3** ($N = 120$, $x \in [1.5R, 3R]$, 8×15), in cui i probes sono collocati parzialmente all'interno della bolla di ricircolazione del cilindro target (Figura 5.6), dove le componenti di velocità passano per valori prossimi allo zero o cambiano segno. La Figura 6.10 mostra la griglia sovrapposta al campo di riferimento: le stazioni più prossime al corpo cadono nella regione di ricircolazione, dove il segnale è instabile. In queste condizioni il denominatore di normalizzazione $U_k^{\text{ref}}(\mathbf{x}_i)$ della funzione **VarianceScaled** si avvicina a zero e l'errore relativo $\varepsilon_{ik} = (U_k - U_k^{\text{ref}})/U_k^{\text{ref}}$ si amplifica in modo abnorme.

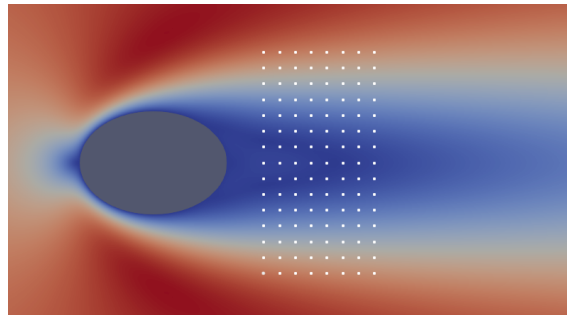


Figura 6.10: Griglia di probes del caso **Probes1'5_3** ($x \in [1.5R, 3R]$, 8×15) sovrapposta al campo di velocità di riferimento. Le stazioni longitudinali più vicine al corpo ($x \lesssim 3R$) si trovano all'interno o al margine della bolla di ricircolazione, dove le componenti di velocità attraversano lo zero e il termine di normalizzazione dell'errore relativo diventa numericamente instabile.

La conseguenza è duplice. Da un lato la storia di convergenza (Figura 6.12) parte da un valore iniziale molto elevato ($f_0 \approx 4 \times 10^4$) e, dopo una discesa rapida, si arresta su un plateau ancora modesto $f_{\text{fin}} \approx 0.5$ (tre ordini di grandezza superiore ai casi in scia intermedia e lontana) senza mai raggiungere una convergenza autentica. Dall'altro, e soprattutto, la geometria ricostruita è gravemente distorta: l'RMSE sale a 0.0906, quasi un ordine di grandezza peggiore del caso migliore, con un errore massimo di 0.22 localizzato sui fianchi ($\theta \approx \pm 57$). Il profilo ottimizzato (Figura 6.11) presenta ondulazioni e rientranze irregolari, tipiche di un'ottimizzazione guidata da un segnale rumoroso: alcuni probes all'interno della bolla forniscono errori relativi enormi e non fisici che dominano il gradiente, spingendo la forma in direzioni prive di senso geometrico.

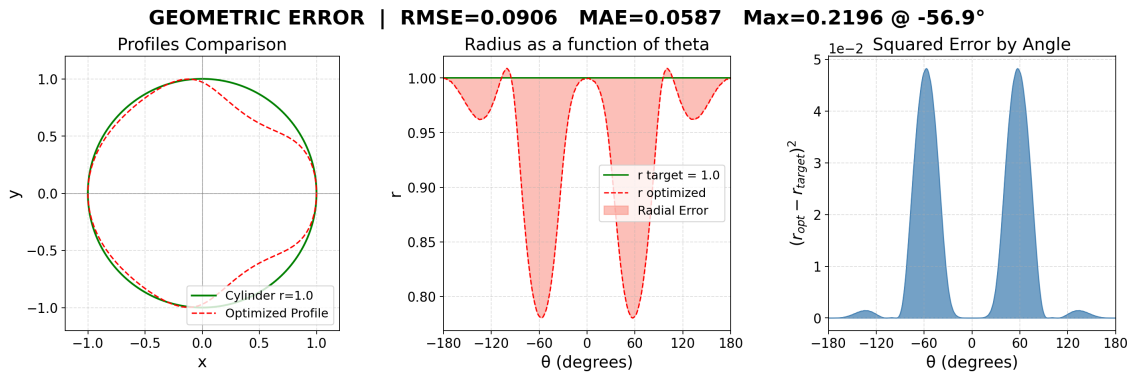


Figura 6.11: Caso Probes1'5_3 ($x \in [1.5R, 3R]$, 120 probes), scia prossima: errore geometrico. Il profilo ottimizzato (rosso tratteggiato) è ondulato e distorto (RMSE = 0.0906, $e_{\text{max}} = 0.22$ a $\theta \approx -57$), lontano dalla circolarità del target. La collocazione dei probes entro la bolla di ricircolazione destabilizza il termine di normalizzazione dell'errore relativo.

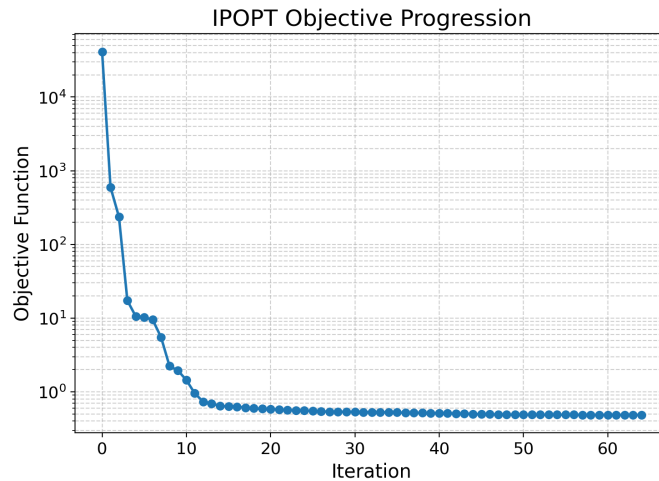


Figura 6.12: Caso Probes1'5_3: storia della funzione obiettivo. Dopo la discesa iniziale la convergenza si arresta su un plateau elevato $f_{\text{fin}} \approx 0.5$, tre ordini di grandezza sopra i casi ben posti, senza mai raggiungere un minimo autentico: la firma quantitativa dell'instabilità numerica nella scia prossima.

6.4.1 Meccanismo dell'instabilità: sensibilità strutturale, inversione di segno e amplificazione

L'origine del fenomeno non risiede, come una prima analisi potrebbe suggerire, nella sola funzione obiettivo, ma nella *sensibilità del campo fluido* all'interno della regione di ricircolazione. Il numero di Reynolds del caso in esame ($Re = 40$, riferito al diametro) è prossimo alla soglia di comparsa del distacco alternato di vortici per il cilindro ($Re_c \approx 47$): il flusso è ancora stazionario, ma l'operatore linearizzato $\partial \mathbf{R} / \partial \mathbf{w}$ è quasi critico, e la sua sensibilità strutturale [la regione in cui una perturbazione localizzata produce la massima risposta, nota in teoria della stabilità come *wavemaker* [30, 31]] è concentrata proprio nella bolla di ricircolazione immediatamente a valle del corpo. I probes collocati in questa zona campionano dunque la regione di massima e più localizzata sensibilità: il campo aggiunto che ne propaga l'informazione verso la forma risulta intenso e spazialmente concentrato su pochi punti (come documentato nella Sezione 7.3), e il gradiente di forma finisce dominato da questi pochi probes anziché dalla struttura complessiva della scia. È un meccanismo *indipendente dalla funzione obiettivo*: l'instabilità permane infatti anche con la formulazione non normalizzata *Variance*, che nella finestra $[2.5R, 4R]$ produce anzi ricostruzioni *peggiori* della *VarianceScaled* (RMSE = 0.073 contro 0.040 a $Re = 40$, e 0.092 contro 0.059 a $Re = 20$; Sezione 6.6), a riprova che la normalizzazione non ne è l'origine.

A questa fragilità strutturale si somma un secondo effetto, geometrico e anch'esso indipendente dall'obiettivo. La bolla di ricircolazione del cilindro target è più estesa di quella della geometria di partenza, l'ellisse, più affusolata: esiste quindi un anello, compreso tra il bordo della bolla iniziale e quello della bolla target, in cui la componente longitudinale del campo di riferimento è negativa ($U_x^{\text{ref}} < 0$, flusso di ritorno del cilindro) mentre quella della forma corrente è ancora positiva ($U_x > 0$). Man mano che l'ottimizzazione fa crescere la bolla, il suo bordo attraversa i probes e la velocità osservata in quei punti passa da positiva a negativa in modo *non monotono* rispetto alle variabili di forma. Poiché l'obiettivo pesa lo scarto al quadrato, $(U_k - U_k^{\text{ref}})^2$, esso è una parabola regolare nella velocità ma, composta con una mappa forma $\rightarrow$ velocità che si *ripiega*, genera un paesaggio di costo non convesso, con minimi locali e direzioni di gradiente poco affidabili. L'elevamento al quadrato, inoltre, non distingue un disaccordo di sola *ampiezza* (velocità dello stesso segno) da un disaccordo *topologico* di segno opposto, cioè presenza contro assenza di flusso di ritorno nel punto, perdendo un'informazione che sarebbe preziosa per ricostruire correttamente l'estensione della bolla: due situazioni fisicamente molto diverse possono contribuire all'obiettivo nella stessa misura.

Su questi due meccanismi di fondo si innesta infine, per la sola *VarianceScaled*, un terzo effetto puramente *amplificatore*. Il suo termine elementare è l'errore relativo $\varepsilon_{ik} = (U_k - U_k^{\text{ref}}) / U_k^{\text{ref}}$, ben condizionato solo dove il denominatore U_k^{ref} è lontano da zero. Sulle linee di separazione e sull'asse di simmetria, dove le componenti del campo di riferimento si annullano, l'errore relativo diverge come $1/U_k^{\text{ref}}$ e, nel termine noto dell'equazione aggiunta (cfr. equazione (5.4)), il peso cresce come $1/(U_k^{\text{ref}})^2$: i pochi probes prossimi a un annullamento del riferimento dominano ulteriormente il gradiente. Ciò spiega perché il caso più profondo nella bolla, Probes1'5_3, sia particolarmente severo con la formulazione normalizzata, con un valore iniziale anomalo dell'obiettivo ($f_0 \approx 4 \times 10^4$, contro $\sim 10^2$ – 10^3 dei casi ben posti). Resta però un'*amplificazione*, non la causa: come mostrato sopra, l'instabilità sopravvive anche in assenza del termine $1/U_k^{\text{ref}}$.

Questa lettura chiarisce anche perché l'infittimento della griglia non risolve il problema. Aggiungere probes all'interno della bolla non attenua la divergenza dei singoli termini patologici; anzi, aumentando il numero di punti collocati vicino agli annullamenti del riferimento, può introdurne di nuovi. Il caso Probes1'5_3 a 450 probes riduce l'RMSE da 0.0906 a 0.0507 solo perché la maggioranza dei probes aggiunti cade in regioni ben condizionate della finestra $[1.5R, 3R]$, che diluiscono statisticamente il contributo dei pochi punti divergenti; ma la ricostruzione resta la peggiore tra tutte le configurazioni con probes esterni alla bolla, e il rapporto RMSE/MAE elevato conferma la presenza di distorsioni localizzate non eliminate. Una soluzione soltanto parziale sarebbe modificare la funzione obiettivo, ad esempio regolarizzando il denominatore con un valore di soglia U_0 ($\varepsilon_{ik} = (U_k - U_k^{\text{ref}}) / \max(|U_k^{\text{ref}}|, U_0)$); tale variante, tuttavia, agirebbe unicamente sull'amplificazione dovuta alla normalizzazione, non sulla sensibilità strutturale né sulla non convessità di fondo, che affliggono anche la formulazione non normalizzata, e introdurrebbe per giunta un parametro arbitrario; è stata quindi evitata a favore della più semplice esclusione della regione $x < 3R$. La ricostruzione ottenuta con 450 probes è riportata in Figura 6.14: l'RMSE scende a 0.0507 ma il profilo resta gravemente distorto e la storia dell'obiettivo mantiene un plateau elevato. Il confronto diretto dei due profili ottimizzati (Figura 6.14) mostra che l'infittimento attenua le ondulazioni senza sanare la distorsione di fondo: il fattore limitante non è la risoluzione spaziale ma la fragilità intrinseca della regione di ricircolazione, cioè la sensibilità strutturale del flusso quasi critico e la non convessità legata all'inversione di segno del campo di riferimento che nessun aumento di sensori può eliminare. Si tratta del comportamento opposto a quello osservato per la finestra intermedia $[2.5R, 4R]$, dove invece infittire la griglia giova (Sezione 6.5.2).

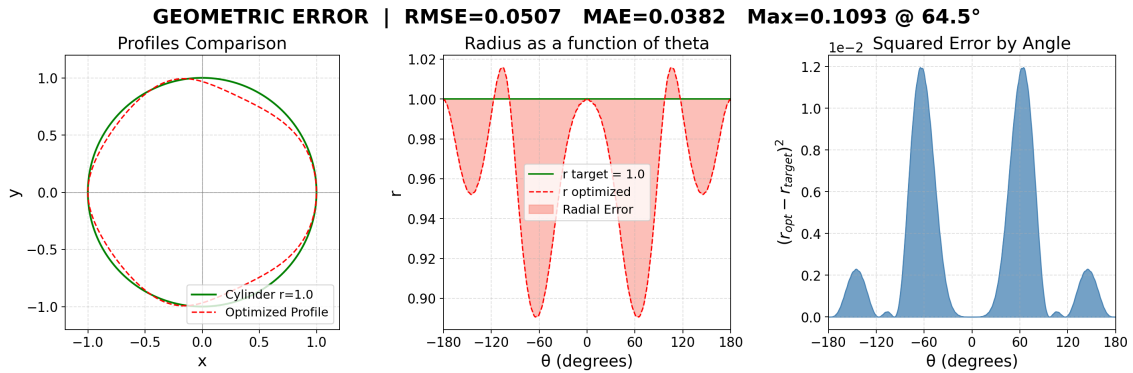


Figura 6.13: Caso Probes1'5_3 infittito a 450 probes (scia prossima). Errore geometrico (RMSE = 0.0507); l'errore si riduce rispetto alla griglia a 120 probes, ma la ricostruzione resta la peggiore tra le configurazioni.

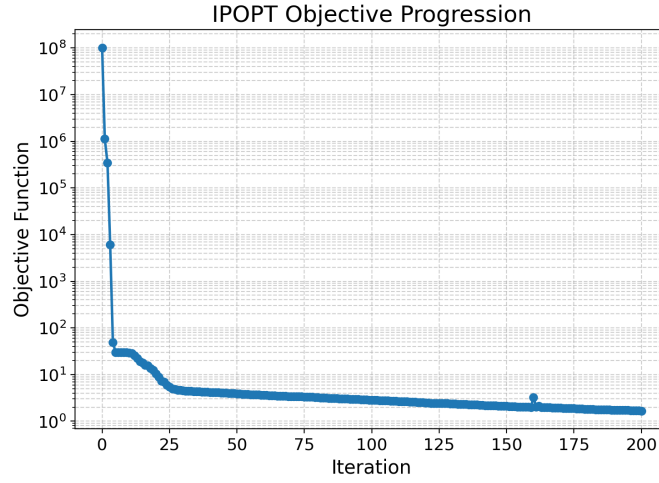


Figura 6.14: Caso *Probes1'5_3* infittito a 450 probes (scia prossima). Storia della funzione obiettivo, ancora caratterizzata da un plateau elevato.

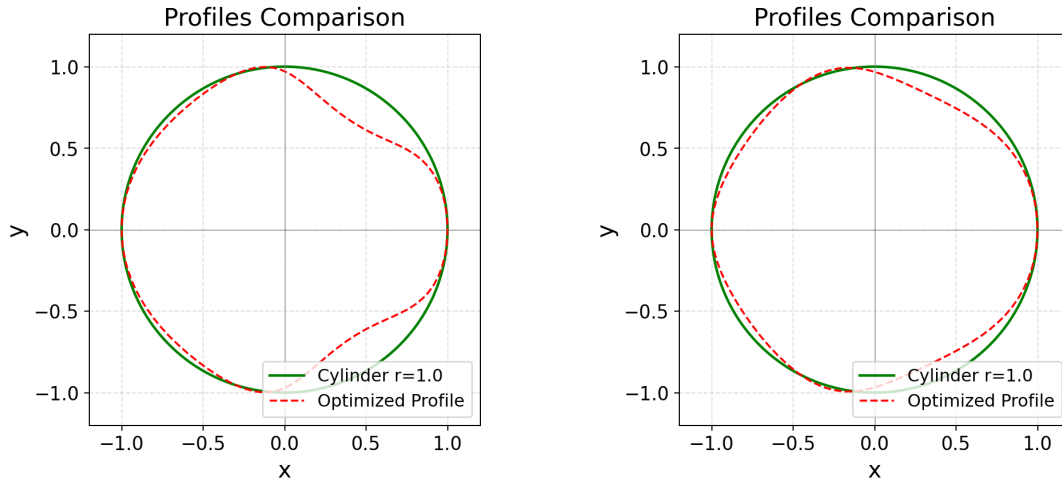


Figura 6.15: Confronto dei profili ottimizzati (rosso tratteggiato sul cilindro target verde) per la scia prossima al variare del numero di probes. Sinistra: 120 probes (RMSE = 0.0906); destra: 450 probes (RMSE = 0.0507). L'infittimento attenua le ondulazioni ma la forma resta visibilmente distorta.

Questo risultato conferma e quantifica la scelta, anticipata nel Capitolo 5, di escludere la regione $x < 3R$ dall'analisi parametrica generale: in quella zona i gradienti aggiuntivi sono dominati dalla sensibilità strutturale della bolla di ricircolazione e dalla non convessità indotta dall'inversione di segno del campo di riferimento indipendentemente dal numero di sensori impiegati e, per la sola *VarianceScaled*, ulteriormente amplificati dalla divergenza del termine di normalizzazione. Il limite inferiore della finestra utile coincide dunque, con buona approssimazione, con l'estensione longitudinale della bolla di ricircolazione del cilindro target ($x \approx 3R$, Figura 5.6); un confine fisico, non convenzionale, del dominio di applicabilità del metodo.

6.5 Analisi di sensibilità al numero di Reynolds

Tutti i casi discussi finora sono a $Re = 40$. Per valutare in che misura il numero di Reynolds influenzi la ricostruibilità della forma, si è ripetuta l'ottimizzazione a $Re = 20$ mantenendo invariata la configurazione dei probes (finestra $x \in [2.5R, 4R]$, $8 \times 15 = 120$ probes, funzione `VarianceScaled` a due componenti). La riduzione del Reynolds si ottiene raddoppiando la viscosità cinematica ($\nu = 0.1$ invece di 0.05), a parità di velocità di freestream e di geometria; il campo di riferimento è rigenerato con la simulazione del cilindro alla nuova viscosità, preservando la natura di *inverse crime* del confronto.

I risultati sono riportati nella Tabella 6.2. Il caso a $Re = 40$ produce una ricostruzione nettamente migliore ($RMSE = 0.0397$) rispetto a quello a $Re = 20$ ($RMSE = 0.0594$), con un miglioramento del 33%. L'interpretazione fisica è diretta: a Reynolds più elevato la scia è più strutturata e la sua firma dipende in modo più marcato dai dettagli della geometria, per cui i probes registrano un segnale più ricco di informazione geometrica; a Reynolds più basso la maggiore diffusione viscosa *leviga* la scia, attenuando la dipendenza del campo dalla forma del corpo e rendendo il problema inverso più ambiguo. Coerentemente, la convergenza a $Re = 20$ è anche più lenta e faticosa: entrambi i casi, avendo probes a partire da $2.5R$ e quindi parzialmente vicini alla bolla, richiedono un elevato numero di iterazioni (> 150) e valori di f_{fin} relativamente alti, ma la ricostruzione a $Re = 40$ resta superiore.

Il confronto diretto delle due ricostruzioni geometriche (Figura 6.16) rende visibile la differenza. A $Re = 20$ (in alto) il profilo ottimizzato si discosta dal cerchio con lobi di errore più profondi e un raggio minimo più basso, mentre a $Re = 40$ (in basso) l'aderenza al target è visibilmente migliore, con lobi di errore più contenuti. La struttura a quattro lobi resta la stessa, segno che il limite della parametrizzazione FFD agisce in entrambi i casi, ma la sua ampiezza è governata dalla ricchezza del segnale di scia, che cresce con il Reynolds.

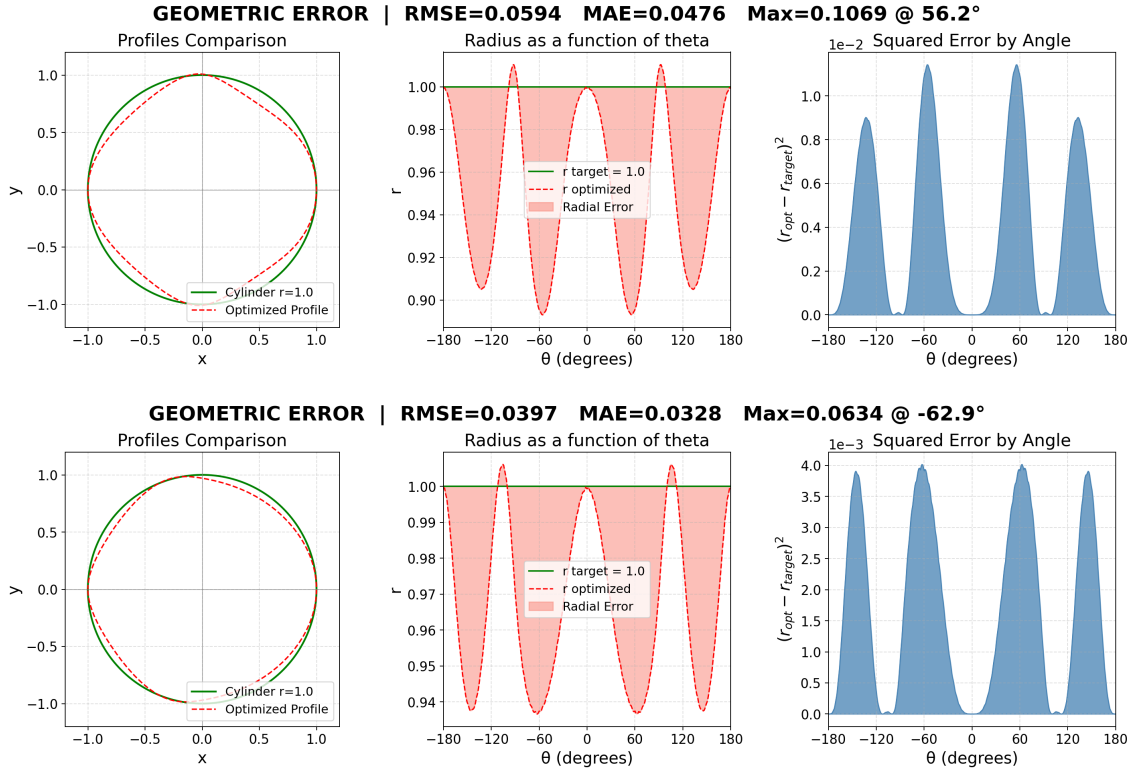


Figura 6.16: Ricostruzione geometrica nella finestra $x \in [2.5R, 4R]$ (8×15 probes) al variare del numero di Reynolds. In alto: $Re = 20$ (RMSE = 0.0594). In basso: $Re = 40$ (RMSE = 0.0397). A Reynolds maggiore la scia meno diffusa porta un segnale geometrico più forte e i lobi di errore si riducono.

Tabella 6.2: Analisi di sensibilità al numero di Reynolds e alla densità dei probes. Configurazione base: finestra $x \in [2.5R, 4R]$, VarianceScaled a due componenti, ellisse $b = 0.7$.

Caso	Re	N	RMSE	MAE	$e_{\max}$
Probes2'5_4 (Re20)	20	120	0.0594	0.0476	0.107
Probes2'5_4 (Re40)	40	120	0.0397	0.0328	0.063
Probes2'5_4 (Re40, densa)	40	~450	0.0237	0.0180	0.051

6.5.1 Interpretazione fisica della dipendenza dal Reynolds

La miglior ricostruibilità a Reynolds più elevato può essere ricondotta al rapporto tra convezione e diffusione che governa la scia. Il numero di Reynolds misura appunto il peso relativo del trasporto convettivo rispetto a quello diffusivo: a Re maggiore la convezione domina, la scia resta “compatta” e trasporta a valle una firma geometrica ben definita per una lunga distanza; a Re minore la diffusione viscosa, più intensa, distribuisce rapidamente la quantità di moto e leviga i gradienti trasversali, dissolvendo prima l’informazione sulla forma. In termini di lunghezze caratteristiche, la scala di diffusione trasversale cresce come $\sqrt{\nu x / U_\infty}$: dimezzando Re (cioè raddoppiando ν) questa scala aumenta di un fattore $\sqrt{2}$, per cui alla stessa distanza x la scia a $Re = 20$ è già più larga e più diffusa di quella a

$Re = 40$, e i probes vi leggono un contrasto geometrico attenuato. La differenza di RMSE (0.0594 contro 0.0397) è la traduzione quantitativa di questa perdita di contrasto.

Coerentemente, anche la struttura della bolla di ricircolazione dipende dal Reynolds: a $Re = 20$ la bolla è più corta di quella a $Re = 40$, il che modifica leggermente il confine inferiore della finestra utile. Per questo motivo la finestra $x \in [2.5R, 4R]$ scelta per il confronto deliberatamente collocata vicino al corpo per massimizzare la sensibilità, si trova nel caso a $Re = 20$ in una posizione relativamente più esterna rispetto alla bolla che nel caso a $Re = 40$; ciò contribuisce a spiegare il numero elevato di iterazioni richiesto in entrambi i casi e la loro convergenza faticosa. Il confronto va quindi inteso a parità di posizione geometrica dei probes, non a parità di posizione relativa rispetto alle strutture di scia, che scalano anch'esse con il Reynolds.

Questi risultati hanno un'implicazione di fattibilità sperimentale. La maggiore ricostruibilità ad alto Reynolds è incoraggiante, poiché le applicazioni pratiche (aerodinamica di veicoli, profili alari, corpi tozzi industriali) operano a Reynolds molto superiori a quelli qui esplorati. Tuttavia, il regime laminare stazionario studiato in questa parte della tesi cessa di valere già poco oltre $Re \approx 47$, dove la scia del cilindro diventa instazionaria e compare il distacco alternato dei vortici (via di Bénard–von Kármán). In regime instazionario o turbolento la firma di scia acquisisce una componente fluttuante che richiede una diversa formulazione dell'obiettivo (basata su medie temporali o su statistiche del campo) e reintroduce il problema della consistenza del modello di turbolenza, discusso nell'introduzione. L'estensione a un profilo alare a Reynolds elevato, con modello di chiusura Spalart–Allmaras, è affrontata nel Capitolo 8 e mostra come la metodologia si adatti a questo regime più complesso.

6.5.2 Effetto della densità dei probes

Sulla stessa configurazione a $Re = 40$ si è inoltre valutato l'effetto di un forte infittimento della griglia di probes (terza riga della Tabella 6.2): triplicando circa il numero di sensori nella medesima finestra, l'RMSE scende ulteriormente da 0.0397 a 0.0237. La Figura 6.17 confronta la griglia standard 8×15 e la variante infittita 15×30 nella medesima finestra $x \in [2.5R, 4R]$.

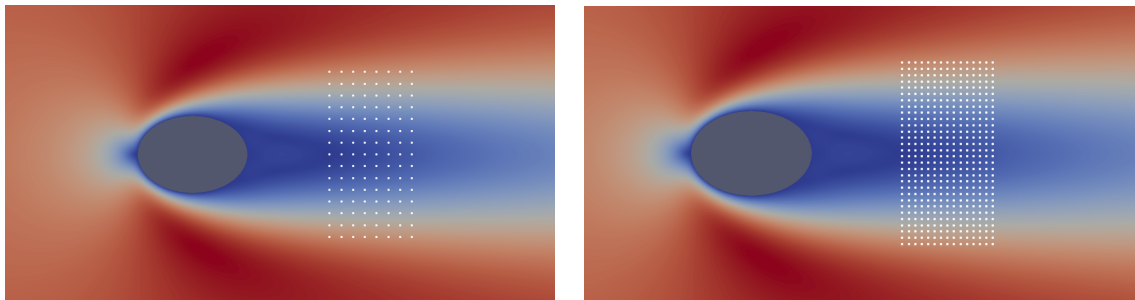


Figura 6.17: Finestra di misura $x \in [2.5R, 4R]$ usata per l'analisi di Reynolds: griglia standard $8 \times 15 = 120$ probes (sinistra) e variante infittita $15 \times 30 \approx 450$ probes (destra). L'infittimento riduce l'RMSE da 0.0397 a 0.0237 perché la finestra, vicina al corpo, campiona un campo a rapida variazione spaziale dove una risoluzione più fine è premiante.

L'infittimento aiuta in questa finestra perché $x \in [2.5R, 4R]$ è ancora relativamente vicina al corpo, dove il campo varia rapidamente e una risoluzione spaziale più fine cattura

gradienti locali significativi, a differenza della scia lontana, dove si è visto che oltre i 120–225 probes il guadagno è marginale.

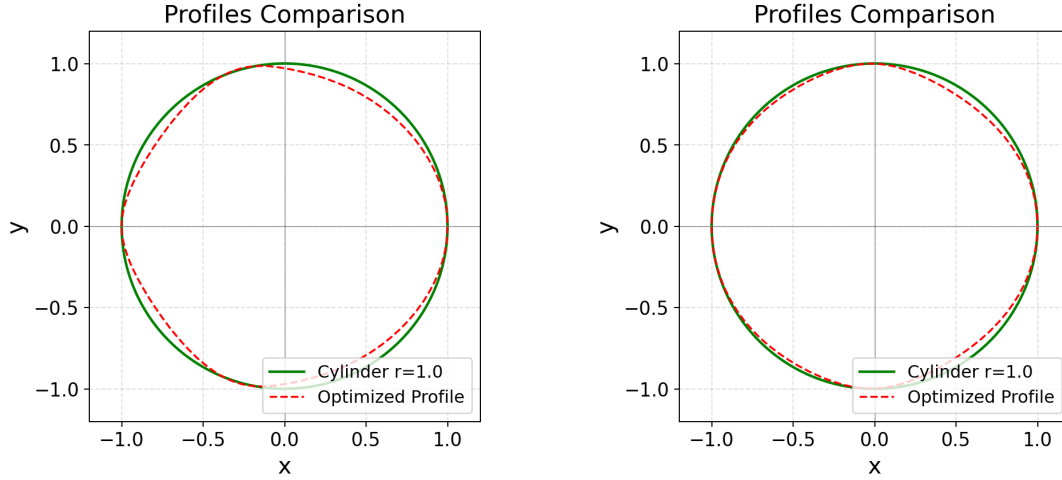


Figura 6.18: Ricostruzione geometrica al variare della densità dei probes nella finestra $x \in [2.5R, 4R]$ a $Re = 40$: profilo ottimizzato (rosso) contro cilindro target (verde). Sinistra: griglia standard $8 \times 15 = 120$ probes (RMSE = 0.0397). Destra: griglia infittita $15 \times 30 \approx 450$ probes (RMSE = 0.0237). L'infittimento avvicina sensibilmente il profilo ricostruito al bersaglio.

Il confronto tra questo risultato e quelli della Sezione 6.3.3 chiarisce che l'effetto del numero di probes non è universale ma dipende dalla *ricchezza spaziale* del campo nella finestra campionata. Dove il campo varia rapidamente su scale piccole come accade vicino al corpo, ai bordi della bolla, o nelle zone di forte gradiente, una risoluzione spaziale fine è premiante, perché ogni probe aggiuntivo campiona una porzione realmente diversa del campo. Dove il campo è liscio e lentamente variabile come nella scia lontana e uniforme, probes vicini forniscono informazione ridondante, e oltre una certa densità l'aggiunta di sensori non riduce ulteriormente l'ambiguità del problema inverso. La densità ottimale dei probes è quindi legata alla scala spaziale locale delle variazioni del campo, non a un valore assoluto: una regola di progetto utile è infittire là dove il campo cambia rapidamente e diradare dove è uniforme.

Questa conclusione trova conferma nel confronto con il caso opposto della scia prossima (Sezione 6.4): là, pur variando il campo rapidamente, infittire la griglia da 120 a 450 probes non migliora la ricostruzione, perché il fattore limitante non è la risoluzione spaziale ma l'instabilità del segnale di riferimento entro la bolla di ricircolazione. La densità dei probes è dunque efficace solo dove il segnale sottostante è ben condizionato, come nella finestra intermedia $[2.5R, 4R]$ trattata qui e inefficace dove il segnale è instabile.

6.6 Confronto della funzione obiettivo: Variance contro VarianceScaled

Come anticipato nella Sezione 2.2.2, la scelta della funzione VarianceScaled che normalizza l'errore puntualmente per il valore locale di riferimento, rispetto alla formulazione classica Variance (normalizzata invece dalla varianza globale del campo) è stata validata

con un confronto sistematico a parità di configurazione di probes. Il confronto è stato condotto su tre posizioni rappresentative (Probes2'5_4, Probes3_6, Probes6_9), rieseguendo per ciascuna l'ottimizzazione con entrambe le formulazioni. I risultati in termini di RMSE geometrico sono raccolti nella Tabella 6.3.

Tabella 6.3: Confronto tra le formulazioni VarianceScaled e Variance a parità di configurazione. RMSE geometrico del profilo ottimizzato ($R = 1$).

Configurazione	Finestra	VarianceScaled	Variance	Migliore
Probes2'5_4 (Re20)	[2.5, 4]R	0.0594	0.0921	Scaled
Probes2'5_4 (Re40)	[2.5, 4]R	0.0397	0.0731	Scaled
Probes3_6	[3, 6]R	0.0166	0.0397	Scaled
Probes6_9	[6, 9]R	0.0054	0.0124	Scaled

Il quadro che emerge dalla Tabella 6.3 è netto: VarianceScaled produce una ricostruzione migliore in *tutte* le configurazioni testate. Il divario è netto già nella finestra intermedia: per Probes3_6 la formulazione Variance classica si arresta a $RMSE = 0.0397$, oltre due volte peggiore dello 0.0166 raggiunto dalla VarianceScaled, pur convergendo in meno iterazioni (circa 27); la ricostruzione risultante è visibilmente meno accurata e sbilanciata (Figura 6.19, in basso), con i fianchi appiattiti e lobi di errore più marcati. Che una convergenza più rapida conduca a una geometria peggiore è un ulteriore esempio del disaccoppiamento tra il valore dell'obiettivo e l'accuratezza geometrica. Lo stesso divario, in misura più contenuta, si osserva nelle finestre a $Re = 20$ e $Re = 40$ con probes a partire da $2.5R$ (rapporti da 1.5 a 1.8 a favore di VarianceScaled) e persino nella scia lontana Probes6_9, dove VarianceScaled (0.0054) resta migliore della Variance classica (0.0124). La ragione della superiorità è che la normalizzazione globale della Variance classica pesa ogni probe in proporzione alla velocità assoluta locale: nelle zone a bassa velocità presenti in abbondanza quando la finestra è vicina o intermedia l'errore viene di fatto ignorato, e l'ottimizzatore corregge preferenzialmente le regioni ad alta velocità, trascurando la struttura della scia e producendo geometrie sbilanciate.

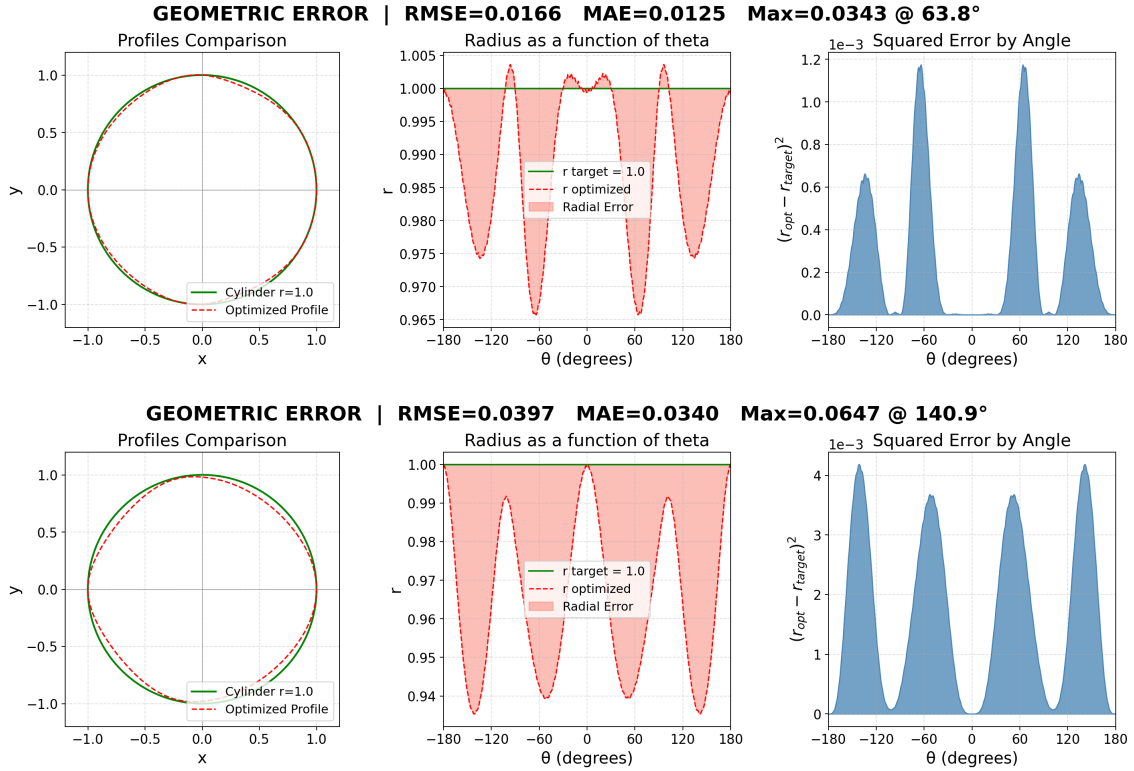


Figura 6.19: Effetto della funzione obiettivo per la finestra intermedia Probes3.6. In alto: *VarianceScaled* (RMSE = 0.0166), ricostruzione accurata. In basso: *Variance classica* (RMSE = 0.0397), ricostruzione meno accurata e sbilanciata: la normalizzazione globale sopprime il contributo dei probes a bassa velocità e concentra la correzione sulle regioni ad alta velocità.

Anche nella scia lontana, dove ci si potrebbe attendere che il vantaggio si annulli, *VarianceScaled* conserva la sua superiorità. Per Probes6_9 essa raggiunge RMSE = 0.0054 contro lo 0.0124 della *Variance classica*: benché in questa regione il campo sia quasi uniforme e la patologia delle basse velocità sia meno pronunciata, la normalizzazione locale continua a distribuire il peso in modo più equilibrato tra i probes, estraendo dal debole segnale residuo un'informazione geometrica più pulita. Il vantaggio, qui pari a un fattore ~ 2.4 , va nella stessa direzione riscontrata in tutte le altre finestre: in nessuna delle configurazioni testate la formulazione classica supera quella normalizzata. Il confronto geometrico diretto delle due ricostruzioni in scia lontana è mostrato in Figura 6.20: entrambi i profili sono prossimi al cilindro, ma quello ottenuto con *VarianceScaled* (in alto) presenta lobi di errore più contenuti e meglio bilanciati rispetto a quello della *Variance classica* (in basso).

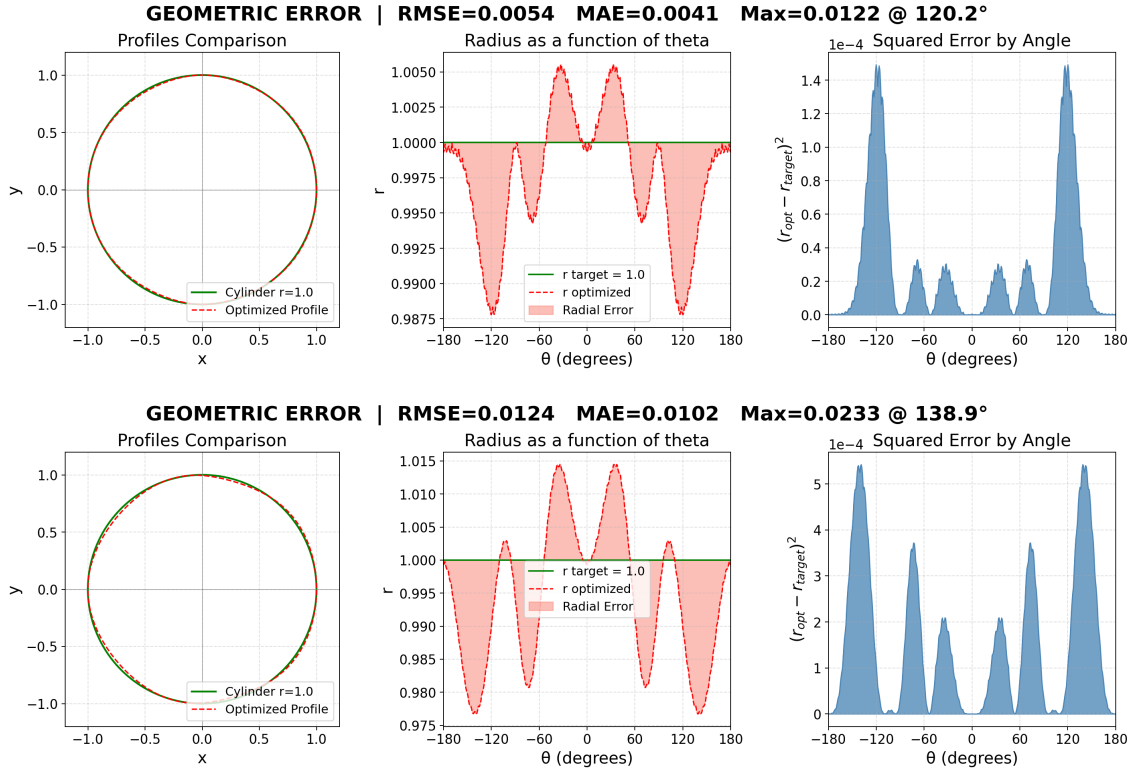


Figura 6.20: Effetto della funzione obiettivo nella finestra di scia lontana *Probes6_9* ($x \in [6R, 9R]$). In alto: *VarianceScaled* (RMSE = 0.0054). In basso: *Variance classica* (RMSE = 0.0124). In scia lontana il campo è privo di zone a bassa velocità critiche e le due formulazioni si avvicinano, ma la normalizzazione locale della *VarianceScaled* produce comunque una ricostruzione più accurata, con lobi di errore meglio bilanciati.

6.6.1 Meccanismo della superiorità di VarianceScaled

La ragione profonda per cui *VarianceScaled* è preferibile risiede nel modo in cui le due formulazioni pesano i probes a bassa velocità, che sono al tempo stesso i più delicati e i più ricchi di informazione geometrica. Nella scia vicina e intermedia le regioni a bassa velocità sono quelle del deficit di scia, la cui depressione di velocità codifica direttamente la forma del corpo. La *Variance classica*, pesando ogni probe in proporzione alla velocità assoluta locale, assegna a queste regioni un peso trascurabile e concentra l'ottimizzazione sui getti laterali ad alta velocità, trascurando la struttura della scia e producendo forme meno accurate e sbilanciate (Figura 6.19). La *VarianceScaled*, normalizzando localmente per il valore di riferimento, ripristina il giusto peso delle zone a bassa velocità ma ad alto contenuto informativo, e ricostruisce correttamente la forma.

Nella scia lontana il deficit di scia si è in gran parte riassorbito e il campo è più uniforme, per cui la disparità di pesi della *Variance classica* è meno severa e le due formulazioni si avvicinano. Tuttavia la debole modulazione del profilo di velocità trasversale ancora legata alla forma vive proprio nelle piccole differenze rispetto al flusso indisturbato, che la normalizzazione locale della *VarianceScaled* valorizza mentre la pesatura assoluta della *Variance* tende a diluire. Da qui il vantaggio, più contenuto ma persistente, di *VarianceScaled* anche a nove raggi dal corpo. È importante osservare che in questa regione i valori di riferimento sono tutti comodamente lontani dallo zero infatti la bolla

di ricircolazione, dove i denominatori si annullano, è ormai molto a monte e perciò la normalizzazione locale opera in condizioni ben poste e non introduce l'instabilità che invece affligge la scia prossima (Sezione 6.4.1). La *VarianceScaled* gode così, nella scia lontana, del meglio di entrambi i mondi: pesatura bilanciata e denominatori ben condizionati.

Va precisato che nella scia lontana il divario è modesto in termini assoluti: 0.0054 e 0.0124 appartengono entrambi alla fascia delle ricostruzioni eccellenti, prossime al limite strutturale imposto dalla parametrizzazione FFD (Sezione 6.2.2). Il punto rilevante non è la misura esatta del vantaggio in quella singola finestra, ma la sua *costanza di segno* attraverso tutte le posizioni esplorate.

Un'ultima osservazione riguarda i valori finali della funzione obiettivo delle due formulazioni, che *non* sono direttamente confrontabili. Nella finestra $[6, 9]R$ la *VarianceScaled* converge a $f_{\text{fin}} = 1.5 \times 10^{-4}$, un valore *superiore* a quello raggiunto dalla *Variance* classica (3.4×10^{-6}), pur producendo una geometria migliore. Ciò è atteso e non contraddittorio: la *VarianceScaled* divide ogni scarto per il valore locale di riferimento, che nella scia è tipicamente compreso tra 0 e 1 (in unità di U_{∞}); dividere per numeri minori dell'unità *amplifica* il valore dell'obiettivo, che risulta quindi sistematicamente più grande di quello della formulazione classica, la quale pesa gli scarti in valore assoluto. Il fatto che la formulazione con l'obiettivo finale più *alto* produca la ricostruzione geometricamente più *accurata* è, tra l'altro, un ulteriore e istruttivo esempio del disaccoppiamento tra obiettivo e qualità geometrica discusso nella Sezione 6.3.2: la magnitudine di f non è mai, di per sé, un indicatore della bontà della ricostruzione, tanto meno quando si confrontano formulazioni con normalizzazioni diverse.

La conclusione metodologica è netta: *VarianceScaled* è preferibile perché è al tempo stesso più accurata in tutte le posizioni testate e più *robusta*: mantiene l'RMSE su valori bassi ovunque, mentre la *Variance* classica, pur competitiva nella scia lontana, degrada sensibilmente quando la finestra include zone a bassa velocità. Poiché lo studio parametrico esplora sistematicamente posizioni vicine, intermedie e lontane, e poiché in un'applicazione reale la posizione ottimale dei sensori non è nota a priori, l'adozione di un'unica funzione obiettivo robusta e uniformemente accurata è largamente preferibile a una scelta caso per caso che rischierebbe un forte degrado in condizioni sfavorevoli. Per questo *VarianceScaled* è stata adottata come formulazione di riferimento per l'intera tesi.

6.7 Casi estremi: probe molto lontane e colonna singola

Per delimitare i confini di applicabilità del metodo si sono infine analizzate due configurazioni ai margini dello spazio di progetto: probes collocate all'estremo del dominio computazionale e una singola colonna di probes. I risultati sono raccolti nella Tabella 6.4, mentre la Figura 6.21 illustra due delle configurazioni più radicali: la griglia a oltre venti raggi dal corpo e la singola colonna trasversale.

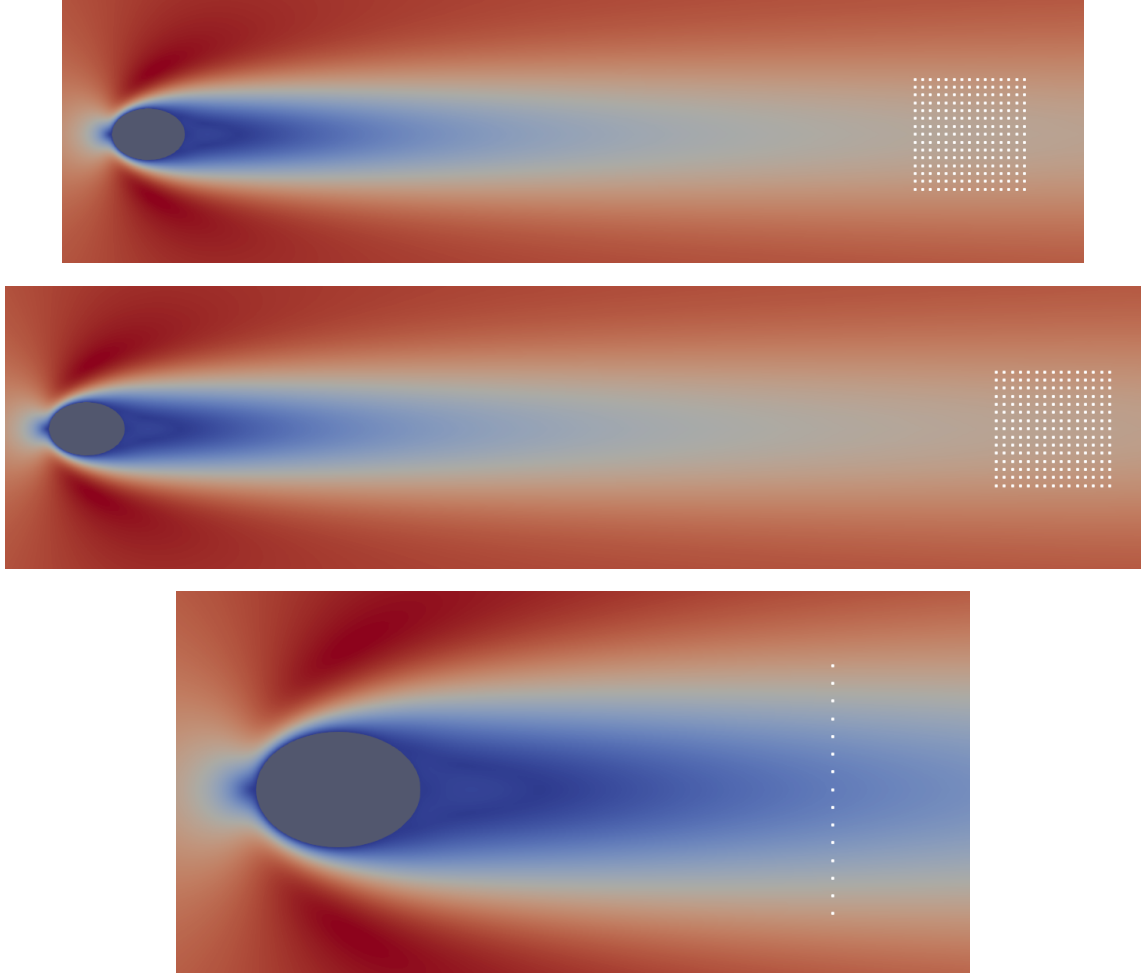


Figura 6.21: Configurazioni estreme. In alto: griglia 15×15 a $x \in [21R, 24R]$, nella parte terminale del dominio ($R_\infty = 30R$); nonostante la distanza, la ricostruzione mantiene $\text{RMSE} = 0.0248$. Al centro: griglia 15×15 ancora più a valle, $x \in [24R, 27R]$, con risultato pressoché identico ($\text{RMSE} = 0.0252$): segno che l'informazione geometrica residua ha raggiunto un asintoto. In basso: singola colonna di 15 probes a $x = 6R$ (Oneprobe_6R), che riduce al minimo l'estensione longitudinale della misura pur ottenendo $\text{RMSE} = 0.0251$.

Tabella 6.4: Casi estremi. Tutti a $Re = 40$, VarianceScaled a una componente, ellisse di partenza $b = 0.7$.

Caso	Finestra	Griglia	Iter	RMSE	MAE	$e_{\max}$
Probes21_24	$[21, 24]R$	15×15	74	0.0248	0.0151	0.060
Probes24_27	$[24, 27]R$	15×15	74	0.0252	0.0155	0.061
Oneprobe_6R	$x = 6R$	1×15	44	0.0251	0.0193	0.046

Probe all'estremo del dominio. Le finestre $x \in [21R, 24R]$ e $x \in [24R, 27R]$ collocano i probes nella parte terminale del dominio computazionale (raggio esterno $R_\infty = 30R$), a oltre venti raggi dal corpo. Sorprendentemente, la ricostruzione rimane buona e stabile: RMSE = 0.0248 e 0.0252 rispettivamente, con storie di convergenza pulite ($f_{\text{fin}} \sim 10^{-6}$ in 74 iterazioni) e profili accurati. Il fatto che i due casi diano risultati quasi identici indica che l'informazione geometrica residua nella scia raggiunge un asintoto: oltre una certa distanza il campo continua a portare una firma debole ma sufficiente a vincolare la forma con un errore di circa il 2.5% del raggio. Questo comportamento, notevole di per sé, è reso possibile dalla natura di *inverse crime* del problema riguardante l'assenza di rumore di misura e coincidenza perfetta dei modelli e non va estrapolato acriticamente al caso sperimentale, dove il rumore renderebbe inutilizzabile una firma così tenue.

Colonna singola di probes. Il caso *Oneprobe_6R* impiega una sola stazione longitudinale a $x = 6R$ (una colonna di 15 probes lungo y), riducendo al minimo l'estensione streamwise della misura. Anche in questa configurazione fortemente impoverita la ricostruzione resta valida (RMSE = 0.0251), confrontabile con i casi a probes molto lontane e con la variante a 60 probes della scia lontana. Ciò mostra che l'informazione trasversale (y) di una singola sezione della scia è già sufficiente a vincolare le cinque variabili di forma FFD: il metodo non richiede necessariamente una griglia estesa in direzione longitudinale, purché la sezione sia collocata al di fuori della bolla di ricircolazione.

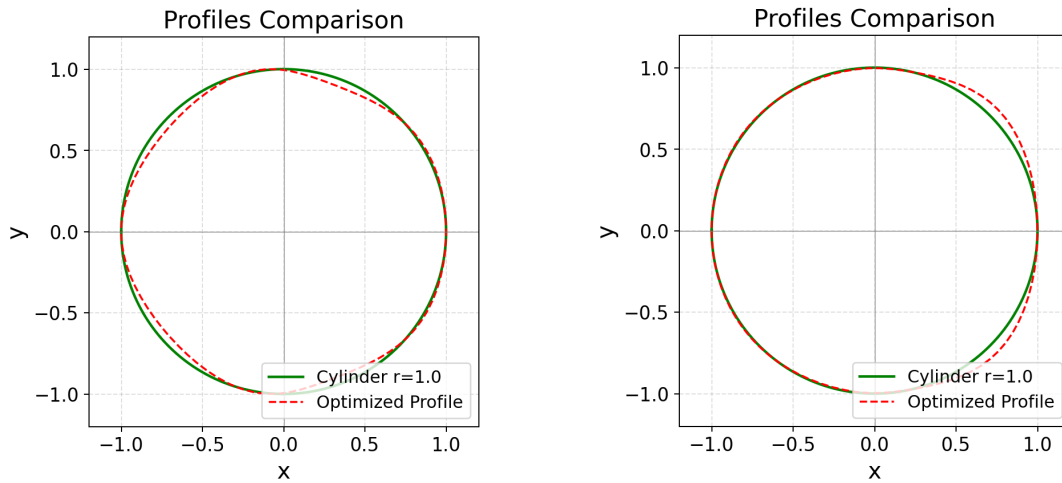


Figura 6.22: Ricostruzione geometrica nei casi estremi: profilo ottimizzato (rosso) contro cilindro target (verde). Sinistra: colonna singola *Oneprobe_6R*, una sola stazione a $x = 6R$ (RMSE = 0.0251). Destra: griglia 15×15 nella parte terminale del dominio, $x \in [24R, 27R]$ (RMSE = 0.0252). Nonostante le condizioni di misura estreme (misura minima l'una, distanza massima l'altra), entrambe ricostruiscono il cilindro con scostamenti contenuti.

6.7.1 Identificabilità: quanti sensori servono davvero

Il risultato della colonna singola merita un approfondimento perché tocca la questione dell'*identificabilità* del problema inverso. Lo spazio di progetto ha dimensione cinque (le cinque stazioni FFD simmetriche); in linea di principio, cinque osservazioni indipendenti sono sufficienti a determinare cinque incognite. Una colonna di 15 probes lungo y fornisce, per ciascuna delle due componenti di velocità, quindici valori largamente ridondanti rispetto ai cinque gradi di libertà, purché tali valori siano sufficientemente indipendenti tra loro e sensibili alle variabili di forma. Il fatto che *Oneprobe_6R* ricostruisca la geometria con $RMSE = 0.0251$, sostanzialmente pari ai casi con centinaia di probes, mostra che il problema è, dal punto di vista dell'informazione, ampiamente sovradeterminato già con una sezione trasversale della scia.

Questo spiega anche perché, oltre una certa soglia, aggiungere probes produca rendimenti decrescenti: una volta che la misura vincola le cinque direzioni dello spazio di progetto, ulteriori sensori aggiungono soprattutto ridondanza, utile a mediare il rumore ma non a ridurre l'ambiguità geometrica di fondo. Il beneficio di griglie ampie non è dunque tanto nel numero di probes in sé, quanto nel fatto che una finestra più estesa campiona regioni fisicamente diverse della scia, con contenuto informativo complementare come discusso nella Sezione 6.3.4, piuttosto che ripetere la stessa informazione. La lezione di progetto è che conta più la *diversità* spaziale della misura che la sua numerosità.

È istruttivo confrontare quantitativamente la colonna singola con le griglie complete alla stessa distanza. La colonna *Oneprobe_6R* (15 probes, una sola stazione a 6R) ottiene $RMSE = 0.0251$; le griglie in scia lontana con la stessa origine longitudinale ottengono 0.0224 ($4 \times 15 = 60$ probes su $[6, 6.7]R$), 0.0130 ($8 \times 15 = 120$ su $[6, 7.5]R$) e 0.0054 ($15 \times 15 = 225$ su $[6, 9]R$). La colonna singola è quindi comparabile alla griglia più piccola (60 probes), ma resta circa cinque volte meno accurata della griglia piena. Il confronto conferma la lettura data sopra: la singola sezione trasversale è sufficiente a *identificare* la forma (cioè a determinare univocamente le cinque variabili FFD), ma l'estensione longitudinale della misura che aggiunge sezioni della scia a distanze diverse, con informazione parzialmente complementare permette di *raffinare* la ricostruzione oltre il livello raggiungibile con una sola stazione. Vale infine la pena notare che *Oneprobe_6R* converge in sole 44 iterazioni con una storia pulita: la povertà della misura non compromette la robustezza dell'ottimizzazione, ma soltanto il livello di accuratezza asintotico.

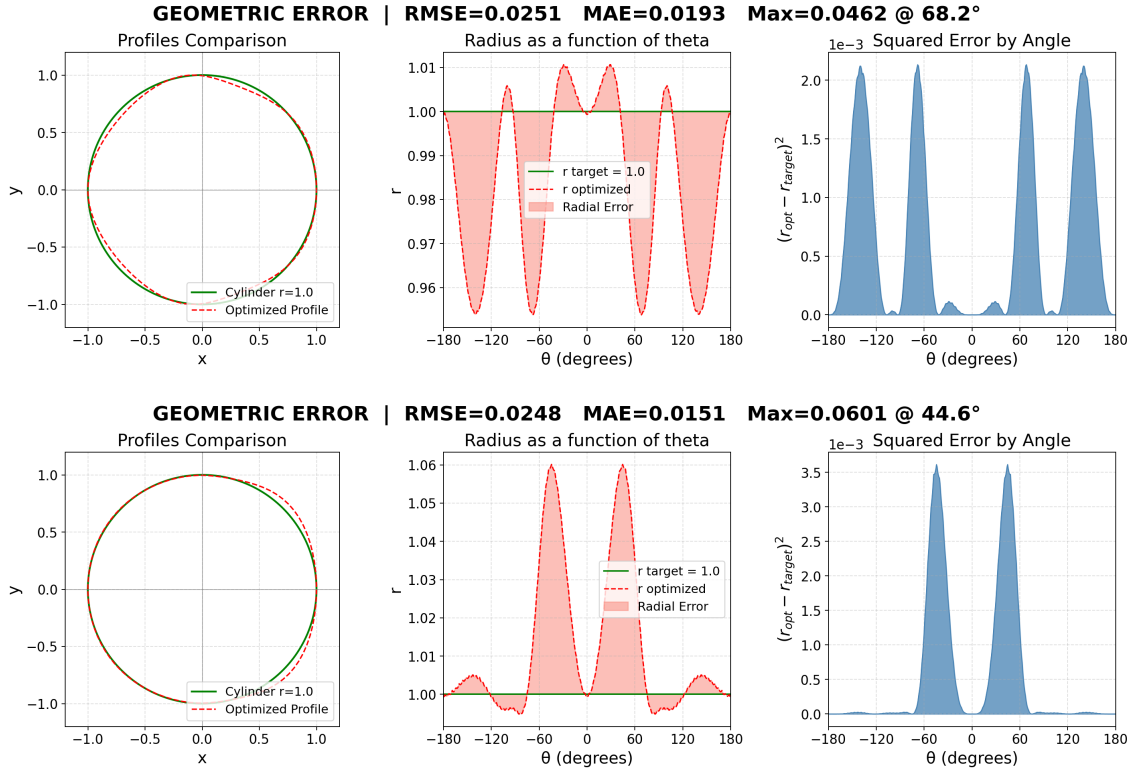


Figura 6.23: Ricostruzioni in configurazioni estreme. In alto: colonna singola *Oneprobe.6R* (una sola stazione a $x = 6R$, $RMSE = 0.0251$). In basso: griglia 15×15 a $x \in [21R, 24R]$ ($RMSE = 0.0248$). Entrambe, pur ai margini opposti dello spazio di progetto (misura minima l'una, distanza massima l'altra) ricostruiscono il cilindro con un errore attorno al 2.5% del raggio, a riprova della robustezza del metodo nel caso ideale privo di rumore.

6.7.2 Il ruolo dell'*inverse crime*: portata e limiti dei risultati estremi

I risultati dei casi estremi, cioè la ricostruzione a oltre venti raggi dal corpo, il successo della colonna singola, sono notevoli e vanno letti alla luce della natura di *inverse crime* consapevole del problema, ripetutamente richiamata in questa tesi. Poiché il campo di riferimento è generato con lo stesso risolutore, la stessa mesh e la stessa fisica del ciclo di ottimizzazione, non esiste alcuna incongruenza di modello né alcun rumore di misura: la firma di scia, per quanto debole, è *esatta*, e l'ottimizzatore può in linea di principio sfruttarne anche le tracce più tenui. È questa idealità a rendere possibile la ricostruzione da probes collocati a $24R$ – $27R$, dove il contrasto geometrico residuo è dell'ordine di frazioni di punto percentuale della velocità di freestream.

In un contesto sperimentale la situazione sarebbe radicalmente diversa. Le misure ottiche (PIV, LDV) e a sonda hanno un rumore caratteristico dell'ordine dell'uno per cento della velocità o superiore; una firma geometrica più debole del livello di rumore è, per definizione, irrecuperabile. I casi estremi non vanno quindi intesi come una promessa che la ricostruzione da scia molto lontana sia praticabile in laboratorio, bensì come una *mappatura del contenuto informativo intrinseco* del campo: essi mostrano fin dove, in assenza di rumore, l'informazione geometrica persiste. La distanza utile in un esperimento reale sarebbe fissata dal punto in cui il contrasto di scia scende sotto la soglia di rumore, ben

prima dei $24R$ qui esplorati. Correlativamente, i casi vicini e intermedi dove il contrasto è più intenso sono quelli che manterrebbero significato anche in presenza di rumore, il che rafforza l'interesse pratico delle finestre attorno a $3R-9R$ individuate come ottimali. L'estensione del metodo a dati rumorosi e a incongruenze di modello, sulla scia delle problematiche discusse nell'introduzione, costituisce la naturale prosecuzione del lavoro.

6.8 Sintesi dell'analisi parametrica

L'analisi condotta in questo capitolo permette di fissare alcune conclusioni operative che guidano la scelta della configurazione di riferimento adottata nel Capitolo 7. Il quadro complessivo dei diciotto casi validi processati è raccolto nella Tabella 6.5, che ordina le configurazioni per RMSE crescente e ne riassume gli attributi principali; essa costituisce il riferimento sintetico dell'intera campagna parametrica.

Tabella 6.5: Quadro riepilogativo dei diciotto casi validi, ordinati per RMSE geometrico crescente. Salvo diversa indicazione: $Re = 40$, funzione `VarianceScaled`. La colonna "Note" segnala le varianti (var: `Variance` classica; 2c: due componenti; densa: griglia infittita).

Caso	Finestra x	N	RMSE	$e_{\max}$	Note
Probes6_9	$[6, 9]R$	225	0.0054	0.012	—
Probes6_9 (var)	$[6, 9]R$	225	0.0124	0.023	var
Probes6_75	$[6, 7.5]R$	120	0.0130	0.028	—
Probes3_6	$[3, 6]R$	225	0.0166	0.034	—
Probes6_67	$[6, 6.7]R$	60	0.0224	0.047	—
Probes2'5_4 (densa)	$[2.5, 4]R$	~ 450	0.0237	0.051	2c, densa
Probes21_24	$[21, 24]R$	225	0.0248	0.060	—
Oneprobe_6R	$x = 6R$	15	0.0251	0.046	1 colonna
Probes24_27	$[24, 27]R$	225	0.0252	0.061	—
Probes3_45	$[3, 4.5]R$	120	0.0279	0.049	—
Probes3_37	$[3, 3.7]R$	60	0.0384	0.066	—
Probes2'5_4 (Re40)	$[2.5, 4]R$	120	0.0397	0.063	2c
Probes3_6 (var)	$[3, 6]R$	225	0.0397	0.065	var
Probes1'5_3 (450)	$[1.5, 3]R$	450	0.0507	0.130	vicino
Probes2'5_4 (Re20)	$[2.5, 4]R$	120	0.0594	0.107	2c, Re20
Probes2'5_4 (Re40,var)	$[2.5, 4]R$	120	0.0731	0.121	var
Probes1'5_3	$[1.5, 3]R$	120	0.0906	0.220	vicino
Probes2'5_4 (Re20,var)	$[2.5, 4]R$	120	0.0921	0.171	var, Re20

La lettura della Tabella 6.5 conferma a colpo d'occhio le tendenze discusse: in cima si collocano le finestre di scia lontana e intermedia con griglie ampie e funzione `VarianceScaled` con la migliore in assoluto che è proprio Probes6_9 (0.0054), mentre in fondo si trovano i casi con probes nella scia prossima e la `Variance` classica applicata a finestre vicine che includono zone a bassa velocità, con RMSE fino a circa 0.09. È significativo che la variante `Variance` della stessa finestra $[6, 9]R$ (0.0124) si collochi comunque nella parte alta della classifica ma dietro la corrispondente `VarianceScaled`: nella scia lontana e uniforme la formulazione classica è competitiva, ma non supera quella normalizzata. Tra i due estremi, l'ampia fascia di configurazioni con $RMSE \in [0.005, 0.040]$ dimostra che il metodo è robusto e che, purché i probes siano collocati fuori dalla bolla e

si adotti una formulazione dell'obiettivo adeguata, la ricostruzione è di buona qualità in un intervallo molto ampio di condizioni.

In primo luogo, la regione utile per il posizionamento dei probes è delimitata inferiormente dalla bolla di ricircolazione: collocare sensori entro $x < 3R$ destabilizza la funzione obiettivo *VarianceScaled* e produce ricostruzioni distorte ($RMSE > 0.09$), indipendentemente dal numero di sensori. Superiormente, invece, non esiste un limite pratico entro il dominio: anche probes a $24R-27R$ ricostruiscono la forma con $RMSE \approx 0.025$.

In secondo luogo, all'interno della regione utile la ricostruzione migliora monotonamente con l'estensione della finestra e il numero di probes, e il guadagno non si esaurisce entro le griglie esplorate: l'ottimo assoluto è la scia lontana a griglia piena (*Probes6_9*, $RMSE = 0.0054$), seguita a distanza dalla scia intermedia (*Probes3_6*, $RMSE = 0.0166$). La configurazione a 120 probes offre un buon compromesso tra costo e accuratezza (0.0130 in scia lontana), ma la griglia piena a 225 conserva un margine di miglioramento significativo, dimostrando che l'estensione longitudinale della misura è il parametro più efficace. Il valore finale della funzione obiettivo non è un indicatore affidabile della qualità geometrica e va sempre affiancato dalla metrica *RMSE*.

In terzo luogo, la ricostruibilità cresce con il numero di Reynolds (l'*RMSE* passa da 0.059 a $Re = 20$ a 0.040 a $Re = 40$ nella stessa finestra), e la funzione *VarianceScaled* si è rivelata la scelta migliore e più robusta su tutto il dominio, superiore alla *Variance* classica in tutte le posizioni testate. Infine, il metodo si è dimostrato capace di operare persino con una singola colonna di sensori.

6.8.1 Linee guida di progetto

Le osservazioni precedenti si condensano in alcune indicazioni operative per la progettazione di una campagna di inverse design basata sulla scia. Il posizionamento dei probes dovrebbe evitare la bolla di ricircolazione (per il cilindro a $Re = 40$, $x < 3R$), il cui bordo esterno costituisce il confine inferiore fisico della regione utile; collocarsi in questa zona, per quanto ricca di segnale, destabilizza l'errore relativo e produce ricostruzioni distorte. All'estremo opposto, non è necessario allontanarsi eccessivamente: la fascia $x \in [6R, 9R]$ offre il miglior compromesso tra qualità della ricostruzione e stabilità della convergenza nel caso ideale, mentre distanze molto maggiori, pur funzionanti in assenza di rumore, non sarebbero praticabili sperimentalmente. Quanto al numero di sensori, una griglia di 120 probes rappresenta un buon punto di lavoro quando il costo è un vincolo, ma la griglia piena a 225 offre un miglioramento ancora sostanziale (in scia lontana l'*RMSE* dimezza abbondantemente, $0.0130 \rightarrow 0.0054$), mentre la riduzione a 60 degrada sensibilmente il risultato; conviene quindi estendere la griglia il più possibile compatibilmente con il costo. Più che il numero assoluto conta la copertura spaziale: è preferibile distribuire i sensori su regioni fisicamente diverse della scia che portano informazione complementare piuttosto che infittirli in una zona ristretta, salvo dove il campo varia rapidamente su scale piccole. Infine, la formulazione dell'obiettivo va scelta per robustezza: la normalizzazione locale della *VarianceScaled* è indispensabile ogni volta che la finestra di misura include regioni a bassa velocità, condizione quasi sempre verificata nelle posizioni di maggior interesse pratico.

È utile sottolineare che queste linee guida sono state ricavate nel contesto specifico del cilindro laminare a $Re = 40$ con parametrizzazione FFD a cinque stazioni, e che le distanze caratteristiche (l'estensione della bolla, la posizione ottimale delle finestre) scalano con il

numero di Reynolds e con la geometria del corpo. Il loro valore non risiede quindi nei numeri assoluti, ma nei principi che esprimono: la regione utile è delimitata inferiormente dalle strutture di ricircolazione e superiormente dal livello di rumore; la qualità della ricostruzione dipende dal contenuto informativo del campo più che dalla numerosità della misura; e la scelta della funzione obiettivo deve garantire robustezza rispetto alla variabilità delle condizioni. Questi principi sono trasferibili a geometrie e regimi diversi, incluso il profilo alare turbolento del Capitolo 8.

Sulla base di questi risultati, il Capitolo 7 approfondisce in dettaglio il caso base, analizzandone la convergenza, i campi fluidodinamici e le tempistiche computazionali.

Capitolo 7

Risultati dell'Inverse Design sul Corpo Tozzo

Il capitolo precedente ha stabilito, attraverso un'ampia campagna comparativa, quali condizioni di misura rendano possibile e accurata la ricostruzione inversa della forma. Questo capitolo cambia prospettiva: anziché confrontare molte configurazioni, si concentra su un unico caso di riferimento e ne analizza in profondità il comportamento, dalla convergenza dell'ottimizzatore alla fisica della sensibilità che guida la deformazione, fino ai costi computazionali. L'obiettivo non è più stabilire *quale* configurazione sia migliore, ma comprendere *come* e *perché* il metodo funzioni, illustrando i meccanismi che sottendono i risultati aggregati del Capitolo 6.

Come caso di riferimento si adotta la configurazione ottimale emersa dall'analisi parametrica: Probes6_9, con griglia piena $15 \times 15 = 225$ probes nella finestra di scia lontana $x \in [6R, 9R]$, funzione obiettivo VarianceScaled a due componenti, ottimizzatore IPOPT, ellisse di partenza $a = 1$, $b = 0.7$ verso il cilindro target di raggio $R = 1$ a $Re = 40$. Questa configurazione ha prodotto la migliore ricostruzione geometrica dell'intero studio (RMSE = 0.0054, appena mezzo punto percentuale del raggio) e converge nel minor numero di iterazioni tra le finestre a griglia piena (56), risultando quindi al tempo stesso la più accurata e la più efficiente: il candidato naturale per un'analisi dettagliata. Dove utile al confronto, si richiamano anche i casi estremi già introdotti — in particolare la scia prossima Probes1'5_3, il cui comportamento patologico illumina per contrasto i meccanismi del caso ottimale.

7.1 Convergenza e ricostruzione geometrica del caso di riferimento

L'ottimizzazione del caso Probes6_9 parte da un'ellisse la cui firma di scia differisce nettamente da quella del cilindro target: il valore iniziale della funzione obiettivo è $f_0 = 1.26 \times 10^2$ (nella normalizzazione con fattore di scala $s = 10$ adottata, cfr. Sezione 5.3.3). Nel corso di 56 iterazioni IPOPT la funzione obiettivo scende fino a $f_{\text{fin}} = 1.52 \times 10^{-4}$, con una riduzione complessiva di quasi sei ordini di grandezza (Figura 7.1, sinistra). La storia di convergenza mostra tre fasi distinte, ricorrenti nei casi ben condizionati: una *discesa rapida* iniziale (iterazioni 0–3) in cui l'obiettivo crolla da 126 a $\sim 7 \times 10^{-3}$, oltre quattro ordini di grandezza, corrispondente alla correzione grossolana della forma dall'ellisse verso una geometria approssimativamente circolare; un *plateau intermedio* (iterazioni 3–11) attorno a $f \sim 10^{-3}$, in cui l'ottimizzatore riorganizza la distribuzione della deformazione tra le cinque stazioni FFD; e una *discesa finale* (iterazioni 11–17) fino al plateau di convergenza a $f \sim 1.5 \times 10^{-4}$, seguita da un lungo tratto asintotico stabile. La rapidità complessiva di 56 iterazioni, meno di qualsiasi altra finestra a griglia piena, conferma che la scia lontana offre un problema ben condizionato, mentre l'accuratezza geometrica finale resta la migliore dell'intero studio.

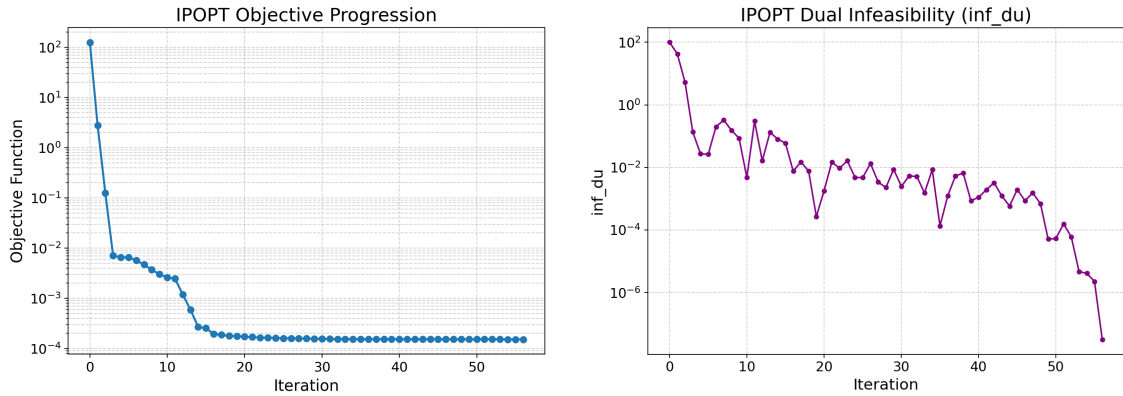


Figura 7.1: Convergenza del caso di riferimento Probes6_9. Sinistra: storia della funzione obiettivo (scala logaritmica), da $f_0 = 126$ a $f_{fin} = 1.5 \times 10^{-4}$ in 56 iterazioni, con discesa rapida iniziale, plateau intermedio e discesa finale. Destra: infeasibility duale inf_du , misura della distanza dal punto stazionario KKT (condizione di ottimalità matematica), che scende di circa dieci ordini di grandezza, da $\sim 10^2$ fino a $\sim 10^{-8}$, confermando il raggiungimento di un minimo autentico e non di un semplice arresto prematuro.

La convergenza della sola funzione obiettivo non basta a garantire il raggiungimento di un ottimo: occorre verificare le condizioni di ottimalità di Karush–Kuhn–Tucker (KKT). A questo scopo si monitora l'*infeasibility duale* inf_du , il residuo della stazionarietà del Lagrangiano. Il suo significato è il seguente: in un problema di ottimizzazione vincolata la condizione necessaria del primo ordine per un minimo (la stazionarietà KKT) richiede che il gradiente del Lagrangiano rispetto alle variabili di progetto si annulli, ovvero che il gradiente della funzione obiettivo sia esattamente bilanciato da quello dei vincoli attivi, pesati dai rispettivi moltiplicatori; geometricamente, ciò equivale all'assenza di una qualsiasi direzione ammissibile lungo cui l'obiettivo possa ancora diminuire. L'infeasibility duale è la norma di questo residuo e quantifica di quanto tale bilancio è violato, ossia quanto il punto corrente è lontano dal soddisfare la condizione di ottimo: un valore elevato segnala che esiste ancora una direzione di discesa ammissibile e che l'ottimizzazione non è conclusa, mentre un valore che tende a zero certifica che nessuna direzione ammissibile riduce ulteriormente l'obiettivo, cioè che il punto è, al primo ordine, un ottimo matematico del problema. È in questo senso che inf_du si traduce nella vicinanza al punto di ottimo matematico indicata nel titolo del grafico. La Figura 7.1 (destra) mostra che inf_du decresce da $\sim 10^2$ fino a valori dell'ordine di 10^{-8} , un calo di circa dieci ordini di grandezza. Questo comportamento certifica che l'ottimizzazione non si è arrestata per esaurimento delle iterazioni o per stallo numerico, ma ha effettivamente individuato un punto stazionario del problema vincolato: il minimo raggiunto è matematicamente robusto. La coesistenza di una funzione obiettivo residua non nulla ($f_{fin} \sim 10^{-4}$) e di un'infeasibility duale prossima allo zero non è contraddittoria: significa che il punto è stazionario (il gradiente della funzione costo è annullato entro le tolleranze), ma che il minimo stesso non è nullo, poiché la parametrizzazione FFD non consente di riprodurre esattamente il cilindro (Sezione 6.2.2).

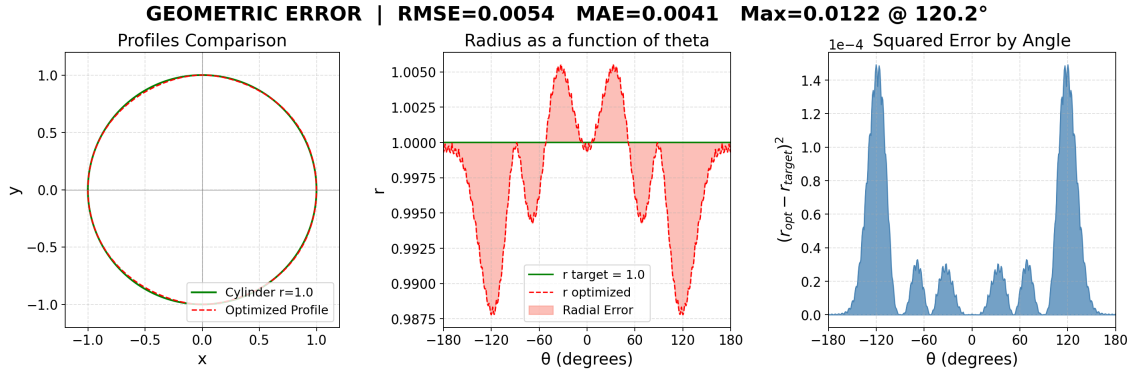


Figura 7.2: Ricostruzione geometrica del caso di riferimento Probes6_9. Sinistra: profilo ottimizzato (rosso tratteggiato) sovrapposto al cilindro target (verde); la coincidenza è quasi perfetta. Centro: raggio $r_{opt}(\theta)$, con scostamenti contenuti entro $\pm 0.012R$. Destra: errore quadratico per angolo, con la firma a quattro lobi a $\theta \approx \pm 35$ e $\theta \approx \pm 120$. Metriche: RMSE = 0.0054, MAE = 0.0041, $e_{max} = 0.012$ a $\theta = 120.2$.

La geometria ricostruita al termine dell'ottimizzazione è mostrata in Figura 7.2. Il profilo ottimizzato aderisce al cilindro target con uno scostamento radiale medio quadratico RMSE = 0.0054, pari a circa mezzo punto percentuale del raggio, e un errore massimo di appena $0.012R$ localizzato nella regione posteriore-laterale ($\theta = 120.2$). La deformazione ha trasformato con successo l'ellisse iniziale, schiacciata del 30% in direzione trasversale ($b = 0.7$), in una forma praticamente circolare, agendo unicamente sulle cinque stazioni FFD guidate dalla firma di scia campionata nove raggi a valle. L'errore residuo, distribuito nei quattro lobi caratteristici già discussi nella Sezione 6.2.2, è di natura strutturale (limite della parametrizzazione) e non riducibile agendo sulla misura: il caso di riferimento è quindi, di fatto, prossimo al miglior risultato ottenibile con la FFD a cinque stazioni adottata.

7.2 Confronto qualitativo dei profili ricostruiti

Il confronto diretto dei profili ottimizzati di configurazioni diverse offre una lettura visiva immediata di come la qualità della ricostruzione dipenda dalle condizioni di misura. La Figura 7.3 contrappone i due estremi dello spettro incontrato nel Capitolo 6: la ricostruzione dalla scia prossima (Probes1'5_3, $x \in [1.5R, 3R]$) e quella da probes molto lontani (Probes21_24, $x \in [21R, 24R]$).

Il contrasto è netto e istruttivo. Il profilo ricostruito dalla scia prossima (Figura 7.3, sinistra), pur partendo dalla regione a più alto contenuto di segnale, è visibilmente distorto: presenta ondulazioni e rientranze irregolari, prive di simmetria circolare, sintomo di un'ottimizzazione guidata da gradienti rumorosi. Il profilo ricostruito da oltre venti raggi di distanza (Figura 7.3, destra), al contrario, è quasi indistinguibile dal cerchio target, con solo una leggera ellitticità residua. Questo confronto che oppone la peggiore alla quasi-migliore delle ricostruzioni riassume visivamente uno dei risultati centrali della tesi: la qualità della ricostruzione non dipende dall'intensità del segnale disponibile (massima nella scia prossima), ma dalla sua stabilità e dalla sua interpretabilità geometrica. Un segnale forte ma numericamente instabile produce una geometria peggiore di un segnale debole ma pulito. La spiegazione fisica di questo apparente paradosso è contenuta nel campo aggiunto, analizzato nella sezione seguente.

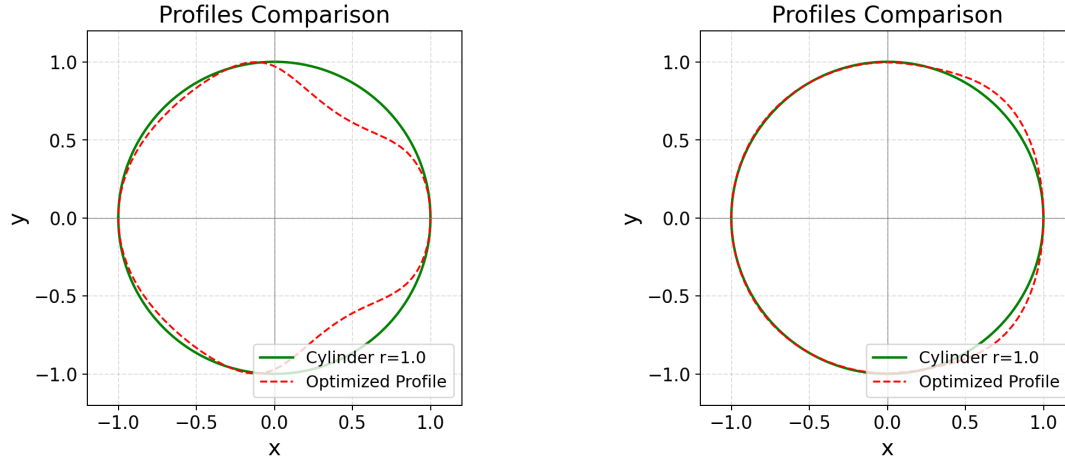


Figura 7.3: Profili ottimizzati (rosso tratteggiato) sovrapposti al cilindro target (verde). Sinistra: scia prossima *Probes1'5_3* ($x \in [1.5R, 3R]$), con profilo ondulato e irregolare (RMSE = 0.0906) prodotto dall'instabilità dell'errore relativo entro la bolla di ricircolazione. Destra: probes molto lontani *Probes21_24* ($x \in [21R, 24R]$), con profilo quasi perfettamente circolare (RMSE = 0.0248) nonostante la distanza estrema dei sensori.

7.3 Il campo aggiunto e la struttura della sensibilità

Il legame tra la firma di scia e la deformazione della forma è mediato dal campo aggiunto ψ , soluzione dell'equazione aggiunta (3.6). Fisicamente, il campo aggiunto rappresenta la *sensibilità* della funzione obiettivo rispetto alle variabili di stato del flusso: in ogni punto del dominio, la sua magnitudine indica quanto una perturbazione locale del campo di velocità influenzerebbe l'obiettivo. Poiché l'obiettivo è definito ai probes, il campo aggiunto è “seminato” in corrispondenza dei probes (attraverso il termine noto $\partial f / \partial \mathbf{U}$ dell'equazione (5.4)) e si propaga da lì verso il resto del dominio risalendo la corrente verso il corpo, poiché l'operatore aggiunto trasporta l'informazione in direzione opposta al flusso primale. È questo trasporto a monte che collega l'errore misurato nella scia alla sensibilità della forma del corpo, chiudendo il ciclo dell'ottimizzazione.

La visualizzazione del campo aggiunto per due configurazioni offre la chiave fisica dei risultati del Capitolo 6. La Figura 7.4 mostra la magnitudine del campo aggiunto (variabile `adjoint_function.U`) per il caso di riferimento in scia lontana (*Probes6_9*) e per il caso patologico in scia prossima (*Probes1'5_3*). La differenza più immediata è di *scala*: la barra dei colori del caso vicino arriva a 1.0, quella del caso lontano a 2×10^{-3} ; una differenza di circa due ordini e mezzo di grandezza nell'intensità della sensibilità.

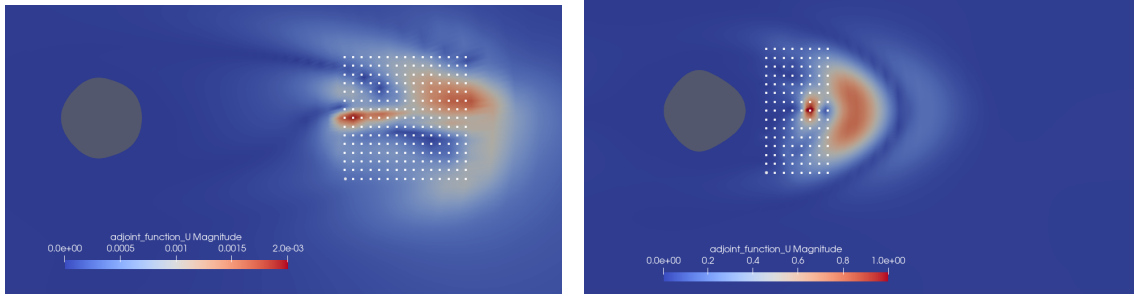


Figura 7.4: *Magnitudine del campo aggiunto `adjoint_function_U` (i punti bianchi indicano i probes). Sinistra: caso di riferimento `Probes6_9` (scia lontana); la sensibilità è debole (scala fino a 2×10^{-3}) e distribuita in modo diffuso, con una tenue traccia che risale verso il corpo. Destra: caso `Probes1'5_3` (scia prossima); la sensibilità è intensa (scala fino a 1.0, circa 500 volte maggiore) e fortemente concentrata attorno ai probes interni alla bolla di ricircolazione. L'intensità estrema e localizzata del caso vicino è la controparte, nello spazio aggiunto, dell'instabilità dell'errore relativo che ne distorce la ricostruzione.*

Questa differenza di scala è la traduzione, nel dominio aggiunto, del meccanismo discusso nella Sezione 6.4.1. Nella scia prossima l'errore relativo $\varepsilon_{ik} = (U_k - U_k^{\text{ref}})/U_k^{\text{ref}}$ e la sua derivata $\partial f / \partial \mathbf{U} \propto 1/(U_k^{\text{ref}})^2$ divergono in prossimità degli annullamenti del campo di riferimento all'interno della bolla; il termine noto dell'equazione aggiunta assume quindi valori enormi in pochi probes patologici, e la soluzione ψ ne eredita l'intensità abnorme e la concentrazione spaziale. Va precisato, alla luce della Sezione 6.4.1, che la *concentrazione* del campo aggiunto nella bolla ha origine strutturale essendo la regione di massima sensibilità del flusso quasi critico, mentre la sua *intensità* estrema è amplificata dal termine $1/(U_k^{\text{ref}})^2$ proprio della sola `VarianceScaled`: con la formulazione non normalizzata la localizzazione permarrebbe, seppure con ampiezza minore. Il gradiente di forma risultante è dominato da questi pochi punti, e spinge la deformazione in direzioni che riducono l'errore locale di quei probes senza riprodurre la struttura complessiva della scia: da qui il profilo distorto della Figura 7.3. Nella scia lontana, al contrario, tutti i denominatori U_k^{ref} sono ben lontani da zero, il termine noto è moderato e uniforme, e il campo aggiunto risulta debole ma *regolare*: la sensibilità è distribuita in modo bilanciato tra i probes, e il gradiente di forma che ne deriva è pulito e informativo, pur essendo di piccola ampiezza. È la regolarità, non l'intensità, a determinare la qualità della ricostruzione.

Si noti infine come, nel caso lontano, la traccia del campo aggiunto che risale verso il corpo sia estremamente tenue: la sensibilità della forma alla misura è debole, coerentemente con il fatto che il campo di velocità a nove raggi porta un'informazione geometrica attenuata. Ciò spiega, dal punto di vista dell'aggiunto, sia la convergenza rapida (il minimo è ampio e facilmente raggiungibile) sia il plateau geometrico osservato nel Capitolo 6 (oltre un certo punto, il gradiente non contiene più informazione utile a rifinire la forma). Il campo aggiunto riconcilia così, in un'unica immagine fisica, i due comportamenti apparentemente contrastanti (bassa funzione obiettivo e buona ma non perfetta geometria) che avevano motivato l'introduzione della metrica RMSE indipendente.

È utile ricordare come il campo aggiunto si traduca concretamente nei gradienti delle variabili di progetto. Il vettore aggiunto ψ , soluzione dell'equazione (3.6), non è il gradiente

finale ma un campo intermedio definito su tutte le celle del dominio; il gradiente rispetto alle cinque variabili di forma FFD si ottiene contraendo ψ con la derivata del residuo rispetto alla geometria, secondo la formula della sensitività $df/d\mathbf{x} = \partial f/\partial \mathbf{x} - \psi^T \partial \mathbf{R}/\partial \mathbf{x}$ (Capitolo 3). Il termine $\partial \mathbf{R}/\partial \mathbf{x}$ è non nullo solo nelle celle prossime alla superficie del corpo, dove la deformazione FFD modifica effettivamente la mesh; di conseguenza, ciò che conta per il gradiente di forma è il *valore del campo aggiunto in prossimità della parete*, non la sua intensità nella scia. La debole traccia che risale fino al corpo nel caso lontano è precisamente ciò che, contratto con $\partial \mathbf{R}/\partial \mathbf{x}$, fornisce i cinque gradienti; la sua tenuità spiega perché i passi dell'ottimizzatore diventino progressivamente più piccoli nella coda di convergenza. Nel caso della scia prossima, invece, l'enorme concentrazione del campo aggiunto vicino al corpo produce gradienti sproporzionati e mal bilanciati tra le cinque stazioni, all'origine della deformazione irregolare osservata. La distinzione tra intensità dell'aggiunto nella scia e sua efficacia sul gradiente di forma alla parete è dunque essenziale per interpretare correttamente le figure.

7.4 Struttura del campo fluidodinamico e della scia

L'analisi del campo aggiunto va completata con quella del campo primale, che ne costituisce il substrato fisico. Il campo di velocità attorno all'ellisse di partenza, mostrato nella Figura 5.2 del Capitolo 5, presenta le caratteristiche tipiche di un flusso laminare stazionario attorno a un corpo tozzo a $Re = 40$: accelerazione laterale in corrispondenza della massima ampiezza del corpo, con velocità che superano quella di freestream sui fianchi, e una scia a valle caratterizzata da una bolla di ricircolazione simmetrica delimitata da due vortici controrotanti (Figura 5.6). La ricostruzione inversa consiste nel modificare la forma del corpo finché questo campo, campionato ai probes, non coincide con quello generato dal cilindro target.

La topologia del flusso è resa in modo particolarmente chiaro dalle due visualizzazioni della Figura 7.5. La rappresentazione LIC (*Line Integral Convolution*) del campo attorno all'ellisse (in alto) mostra la struttura delle linee di corrente in prossimità del corpo: il punto di ristagno anteriore, l'accelerazione lungo i fianchi e, soprattutto, i due centri di ricircolazione simmetrici immediatamente a valle, con le rispettive linee di separazione e di riattacco che delimitano la bolla. La rappresentazione del campo di velocità del cilindro target (in basso), colorata secondo la magnitudine di $\mathbf{U}$ e sovrapposta alle linee di corrente, evidenzia invece la regione di ricircolazione come una zona a velocità molto bassa o negativa (in azzurro, con $U_x < 0$ nel cuore dei vortici) che si estende per circa tre raggi a valle prima del punto di riattacco sull'asse. È esattamente in questa regione, dove la velocità attraversa lo zero, che l'errore relativo della funzione *VarianceScaled* diventa instabile (Sezione 6.4.1); ed è appena all'esterno di essa che si collocano le finestre di misura ottimali.

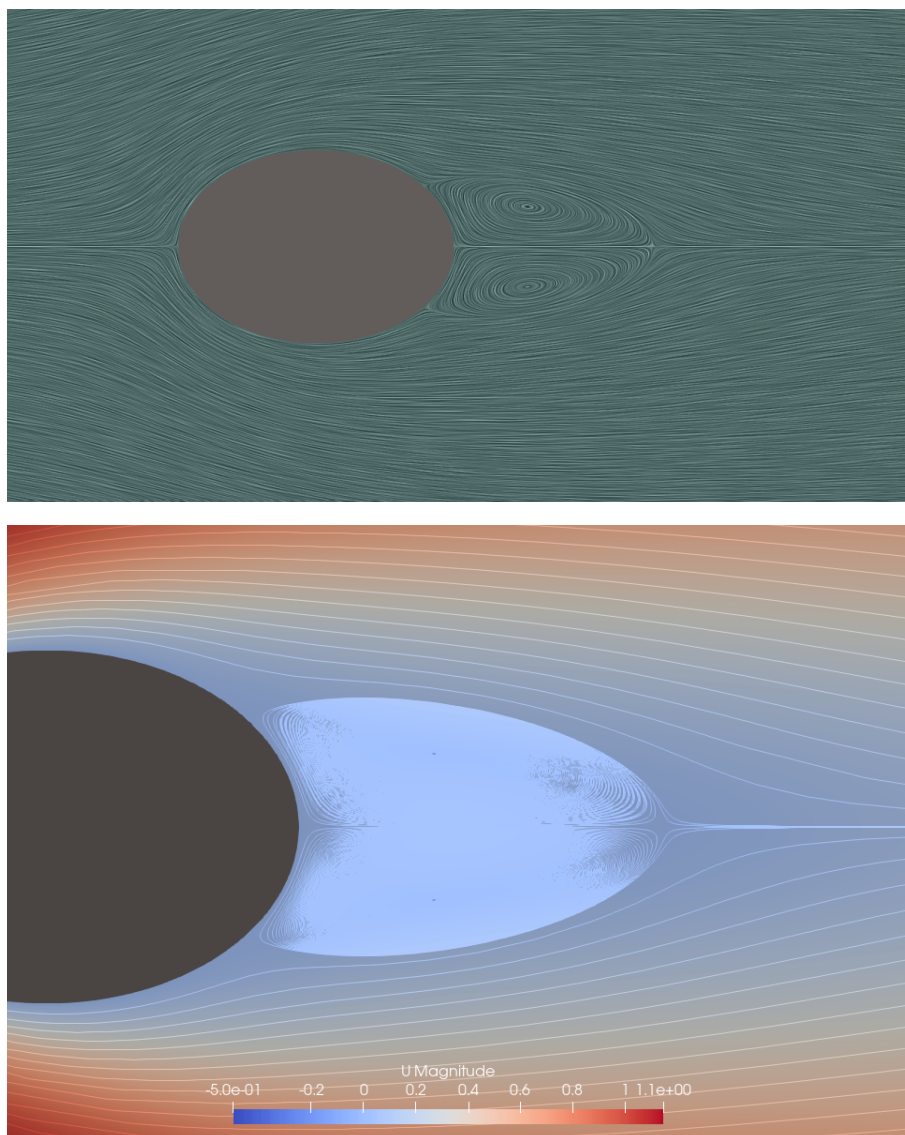


Figura 7.5: Struttura del campo di moto a $Re = 40$. In alto: visualizzazione LIC del campo attorno all'ellisse di partenza; sono ben visibili i due centri di ricircolazione controrotanti nella scia prossima, con le linee di separazione e riattacco. In basso: campo di velocità (magnitudine, con linee di corrente) del cilindro target; la regione azzurra dietro il corpo è la bolla di ricircolazione, dove U_x diventa negativa e il campo attraversa lo zero, cioè la zona in cui l'errore relativo della funzione obiettivo è numericamente instabile.

Il passaggio dall'ellisse al cilindro comporta modifiche fisiche precise del campo di scia, che i probes devono essere in grado di rilevare. Il cilindro, avendo una sezione trasversale maggiore dell'ellisse ($b = 0.7 \rightarrow R = 1$, un aumento del 43% dell'ampiezza trasversale), genera una scia più larga, una bolla di ricircolazione più estesa e un deficit di velocità più marcato lungo l'asse. Sono proprio queste differenze [larghezza della scia, lunghezza della bolla, intensità del deficit] a costituire la firma geometrica che distingue le due forme. Nella scia intermedia queste differenze sono ancora nette e localizzate, e i probes vi leggono un contrasto forte; nella scia lontana esse si sono in gran parte attenuate per diffusione, sopravvivendo solo come una debole modulazione del profilo di velocità trasversale, che il

metodo aggiunto riesce comunque a sfruttare grazie all'assenza di rumore (Sezione 6.7.2). La struttura della bolla di ricircolazione merita un'ultima osservazione, poiché essa gioca un ruolo doppio e opposto nel problema. Da un lato, la bolla è la regione a più alto contenuto di informazione geometrica, perché la sua forma e la sua estensione dipendono sensibilmente dalla geometria del corpo; dall'altro, è la regione in cui il campo di velocità attraversa lo zero, rendendo instabile l'errore relativo e inutilizzabile la misura diretta (Sezione 6.4). Questa dualità di massima informazione dove vi è massima instabilità è la tensione fondamentale del problema di inverse design basato sulla scia, e la scelta di posizionare i probes appena all'esterno della bolla rappresenta il compromesso che la risolve.

7.4.1 Convergenza del campo di errore di velocità

Una visualizzazione particolarmente eloquente del funzionamento del metodo si ottiene mappando sull'intero dominio il campo dell'errore assoluto di velocità $|U_x - U_x^{\text{ref}}|$, cioè la differenza puntuale tra la componente longitudinale simulata e quella del campo di riferimento del cilindro. La Figura 7.6 confronta questo campo alla prima e all'ultima (56^a) iterazione dell'ottimizzazione del caso di riferimento Probes6_9. All'iterazione iniziale (in alto) l'errore è elevato in tutta la scia, e in particolare nella regione occupata dai probes, dove raggiunge valori dell'ordine di 0.1–0.3: è la firma della differenza tra la scia dell'ellisse di partenza e quella del cilindro target. All'iterazione finale (in basso) l'errore nella regione dei probes è sceso a valori prossimi allo zero (blu uniforme): l'ottimizzazione ha portato con successo il campo simulato a coincidere con quello di riferimento *là dove è misurato*, coerentemente con la riduzione della funzione obiettivo di quasi sei ordini di grandezza documentata nella Sezione 7.1.

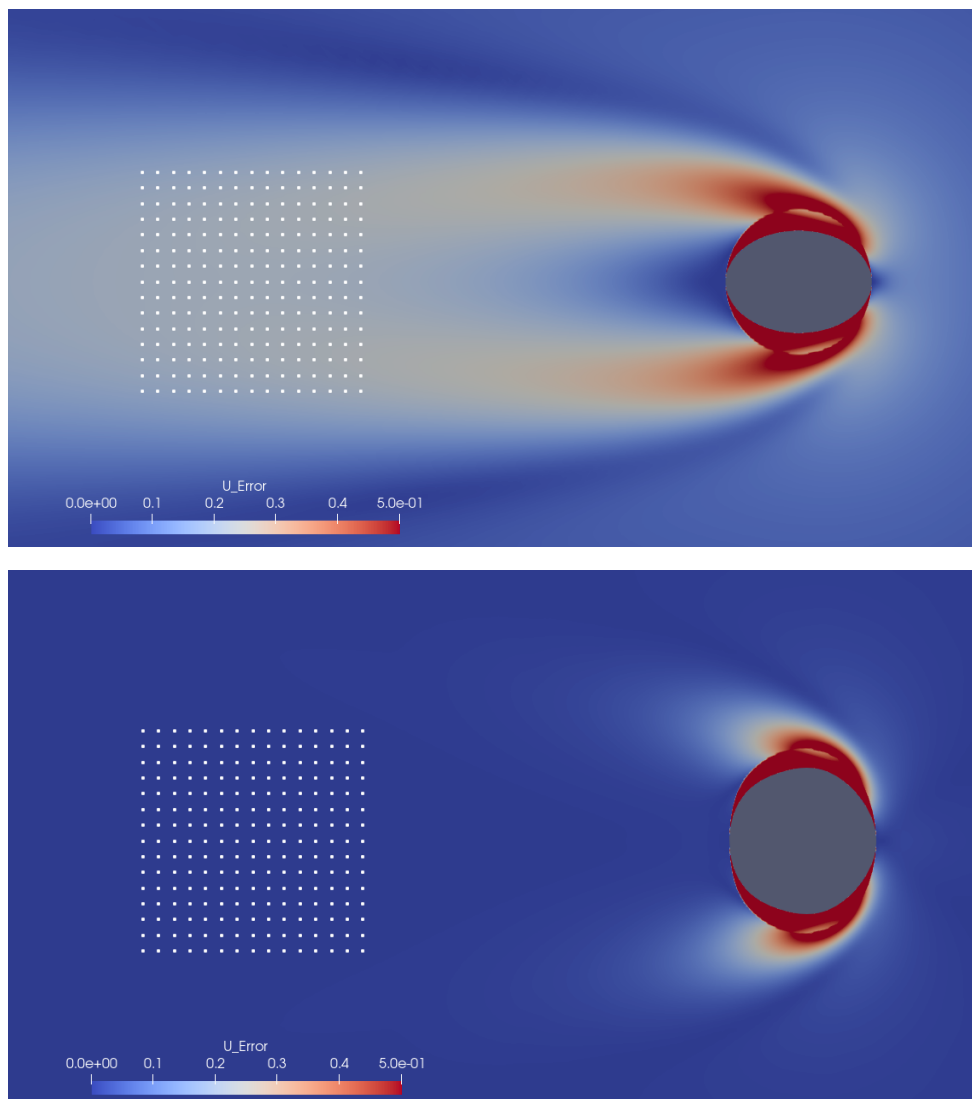


Figura 7.6: Campo dell'errore assoluto di velocità $|U_x - U_x^{ref}|$ per il caso di riferimento Probes6_9 (corpo a destra, regione dei probes a sinistra). In alto: prima iterazione; l'errore è elevato in tutta la scia, inclusa la regione dei probes. In basso: 56^a iterazione; nella regione dei probes l'errore è ormai prossimo allo zero, mentre permane un anello di errore intenso a ridosso della superficie del corpo. Quest'ultimo è un residuo strutturale, esterno alla finestra di misura, che non entra nella funzione obiettivo.

Rimane invece, a tutte le iterazioni, un intenso anello di errore a ridosso della superficie del corpo, chiaramente visibile in entrambi i pannelli. La sua origine è puramente geometrica e non è eliminabile. Il campo di riferimento è quello del cilindro di raggio $R = 1$, all'interno del quale la velocità è nulla (il corpo è solido); l'ellisse di partenza, più schiacciata ($b = 0.7$), lascia invece scoperta una regione anulare che per il cilindro è interna al corpo ma per l'ellisse è esterna. Ad esempio, nel punto $(x, y) = (0, 0.8)$ ci si trova *dentro* il cilindro dove $U^{ref} = 0$, ma contemporaneamente *fuori* dall'ellisse, dove il flusso è non nullo e anzi, per l'accelerazione laterale, può superare la velocità di freestream. In questa fascia anulare la differenza tra i due campi è dell'ordine dell'intera velocità di riferimento, e vi si concentra l'errore massimo. Poiché la topologia della

mesh è fissa e le celle vengono soltanto traslate durante la deformazione FFD, questa discrepanza non può essere sanata nemmeno quando il profilo ha ormai raggiunto la forma circolare: essa persiste identica fino alla 56^a iterazione. È tuttavia importante sottolineare che questo residuo *non compromette in alcun modo il risultato*: la funzione obiettivo campiona la velocità unicamente ai probes, collocati nella scia lontana e ben esterni alla fascia critica, dove l'errore converge regolarmente a zero. L'anello di errore superficiale è quindi un artefatto della visualizzazione a campo intero e della procedura di mappatura del riferimento (mapFields, Sezione 5.3), non un limite della ricostruzione geometrica, che come si è visto raggiunge un'accuratezza di circa mezzo punto percentuale del raggio.

7.5 Residui, convergenza dell'aggiunto e costi computazionali

Ogni iterazione dell'ottimizzazione richiede la soluzione di due problemi accoppiati: il problema primale (le equazioni di Navier–Stokes, risolte da DASSimpleFoam fino alla tolleranza stringente $\|\mathbf{R}\|_\infty < 10^{-13}$) e il problema aggiunto (il sistema lineare (3.8), risolto tramite GMRES matrix-free). Il costo computazionale del caso di riferimento riflette questa struttura. L'ottimizzazione completa di Probes6_9, che richiede 56 iterazioni, ha comportato un tempo di calcolo (*wall-clock*) di circa 9.8×10^3 secondi, ossia poco meno di tre ore, sulla configurazione parallela MPI adottata. La ripartizione di questo tempo è indicativa della natura del problema: circa il 98.5% ($\sim 9.65 \times 10^3$ s) è impiegato nelle valutazioni delle funzioni, cioè nelle soluzioni primale e aggiunta della CFD, mentre solo l'1.5% (~ 147 s) è speso dall'algoritmo di ottimizzazione IPOPT vero e proprio.

Questo squilibrio ha una conseguenza pratica importante: il costo dell'inverse design è dominato interamente dal costo della CFD, e l'overhead dell'ottimizzatore è trascurabile. Ne discende che il numero di iterazioni necessarie alla convergenza è il fattore determinante del costo complessivo, poiché ogni iterazione comporta un numero pressoché costante di soluzioni CFD sulla medesima mesh. È questo che conferisce rilevanza pratica ai risultati sulla velocità di convergenza del Capitolo 6: la Tabella 7.1 traduce i conteggi di iterazione di alcuni casi rappresentativi in un costo relativo, assumendo un costo per iterazione approssimativamente costante.

Tabella 7.1: Costo computazionale relativo di alcuni casi rappresentativi, stimato dal numero di iterazioni IPOPT (il costo per iterazione è approssimativamente costante, essendo dominato dalle soluzioni CFD sulla stessa mesh). Il costo assoluto è misurato solo per il caso di riferimento; gli altri sono stime relative.

Caso	Iterazioni	Costo relativo	RMSE
Probes6_9 (rif.)	56	1.0 (~ 2.7 h)	0.0054
Probes6_75	44	~ 0.79	0.0130
Probes3_6	65	~ 1.16	0.0166
Probes1'5_3	64	~ 1.14	0.0906

La lettura della Tabella 7.1 mostra che, per il caso in esame, accuratezza ed efficienza sono largamente allineate. Il caso di riferimento Probes6_9 è il più accurato dello studio (RMSE = 0.0054) e converge in 56 iterazioni, meno del caso intermedio a griglia piena Probes3_6 (65) e del caso in scia prossima Probes1'5_3 (64): la scia lontana è dunque,

a parità di griglia, sia la più accurata sia tra le più economiche. La configurazione a 120 probes Probes6_75 è ancora più rapida (44 iterazioni) e mantiene un'accuratezza elevata (RMSE = 0.0130): rappresenta il miglior compromesso quando il costo è un vincolo stringente. Il caso in scia prossima Probes1'5_3, invece, cumula gli svantaggi: costa quasi quanto il caso di riferimento (64 iterazioni) ma produce la ricostruzione peggiore (RMSE = 0.0906), sprecando risorse senza alcun beneficio. La scelta della griglia piena in scia lontana come configurazione di riferimento è quindi giustificata su entrambi i fronti della qualità della ricostruzione e dell'efficienza computazionale; un allineamento non scontato che la rende particolarmente adatta come base per le applicazioni più complesse. Va infine osservato che il costo per iterazione è a sua volta dominato dalla tolleranza primale estremamente stringente (10^{-13}) imposta per garantire l'accuratezza delle sensibilità aggiuntive (Sezione 5.1). Un rilassamento di tale tolleranza ridurrebbe il costo di ogni valutazione, ma degraderebbe la precisione del gradiente e, potenzialmente, la convergenza dell'ottimizzazione: si tratta di un ulteriore compromesso, tra costo per iterazione e qualità del gradiente, che nel presente lavoro è stato risolto a favore dell'accuratezza, essendo il costo assoluto (poche ore) pienamente sostenibile per il caso bidimensionale in esame.

7.6 Sintesi

L'analisi approfondita del caso di riferimento ha permesso di ricomporre in un quadro coerente i risultati aggregati del Capitolo 6. La configurazione ottimale in scia lontana converge in modo rapido e robusto: 56 iterazioni, con la funzione obiettivo che cala di quasi sei ordini di grandezza e l'infeasibility duale che certifica il raggiungimento di un punto stazionario KKT, e ricostruisce il cilindro target con un errore radiale medio di circa mezzo punto percentuale, prossimo al limite strutturale della parametrizzazione FFD. Il confronto dei profili e, soprattutto, l'analisi del campo aggiunto hanno fornito la chiave fisica dei comportamenti osservati: la qualità della ricostruzione è governata dalla *regolarità* della sensibilità, non dalla sua intensità, e l'instabilità della scia prossima si manifesta come una concentrazione abnorme del campo aggiunto attorno a pochi probes patologici. Infine, l'analisi dei costi ha mostrato che il carico computazionale è dominato al 98.5% dalle soluzioni CFD, e che nel caso in esame la massima accuratezza e l'efficienza computazionale si realizzano nella stessa configurazione: la griglia piena in scia lontana. Il metodo di inverse design basato sull'aggiunto discreto, validato e caratterizzato in dettaglio sul corpo tozzo bidimensionale, è così pronto per essere applicato a un problema di complessità superiore. Il Capitolo 8 ne affronta l'estensione a un profilo alare a numero di Reynolds elevato e in regime turbolento, dove la firma di scia acquisisce nuova ricchezza e complessità e dove la consistenza del modello di chiusura assume un ruolo centrale.

Capitolo 8

Applicazione Avanzata – Inverse Design di un Profilo Alare

Nei capitoli precedenti la metodologia di inverse design basata sull'aggiunto discreto è stata sviluppata e caratterizzata in dettaglio su un corpo tozzo bidimensionale in regime laminare (deformazione ellisse $\rightarrow$ cilindro a $Re = 40$). Questo capitolo ne verifica la trasferibilità a un problema di complessità superiore e di diretto interesse aerodinamico: la ricostruzione della forma di un profilo alare NACA0012 a partire dalla sua firma di scia, in regime turbolento. L'estensione introduce tre elementi nuovi rispetto al caso tozzo: un numero di Reynolds molto più elevato ($Re = 5 \times 10^4$), un modello di chiusura della turbolenza (Spalart–Allmaras), e una geometria la cui scia è sottile e attaccata anziché dominata da una bolla di ricircolazione. Il filo conduttore trasposto dal benchmark del corpo tozzo al caso del profilo alare resta lo stesso: quale finestra di osservazione dei probes e quale geometria di partenza garantiscono la ricostruzione migliore.

8.1 Introduzione

La determinazione inversa della forma aerodinamica consiste nel recuperare la geometria di un corpo a partire da grandezze di flusso osservate, tipicamente il campo di velocità nella sua scia. Questa classe di problemi è intrinsecamente mal posta: geometrie diverse possono produrre firme di scia simili, e la forma ricostruita è sensibile sia al modello numerico sia alla configurazione di osservazione. Quando il modello diretto si basa sulle equazioni di Navier–Stokes mediate alla Reynolds (RANS), la geometria dedotta dipende fortemente dall'accuratezza della chiusura turbolenta e dalla consistenza delle sensitività aggiuntive.

Uno studio parallelo, basato su un dataset indipendente ad alta fedeltà (LES), ha deliberatamente introdotto una discrepanza di modello, mostrando che l'inconsistenza della chiusura domina la forma ricostruita. Il presente lavoro adotta la posizione opposta e complementare: la chiusura turbolenta è mantenuta fissa e consistente. Il modello di Spalart–Allmaras (S–A) è impiegato sia per generare il campo di riferimento, una simulazione RANS stazionaria del NACA0012 target, sia per guidare l'ottimizzazione inversa. Poiché i dati sono prodotti dallo stesso modello usato nel ciclo aggiunto, il problema è consistente per costruzione. Questo *inverse crime* intenzionale elimina il bias di modello, lasciando una base pulita in cui l'errore residuo di ricostruzione riflette esclusivamente il contenuto informativo delle osservazioni: cioè dove sono collocati i probes e da quale geometria parte l'ottimizzatore.

L'obiettivo dello studio è individuare quale geometria di partenza produca il recupero più accurato e robusto in funzione delle diverse finestre di osservazione. A questo scopo si confrontano tre profili di partenza: NACA0008, NACA0015 e NACA2412, che coprono un intervallo di spessori e di camber, e quindi di distanze geometriche dal target. La metodologia riutilizza il protocollo di traslazione della finestra di probes sviluppato sul benchmark ellisse $\rightarrow$ cilindro, quantificandone il comportamento analogo per la scia attaccata di un profilo alare.

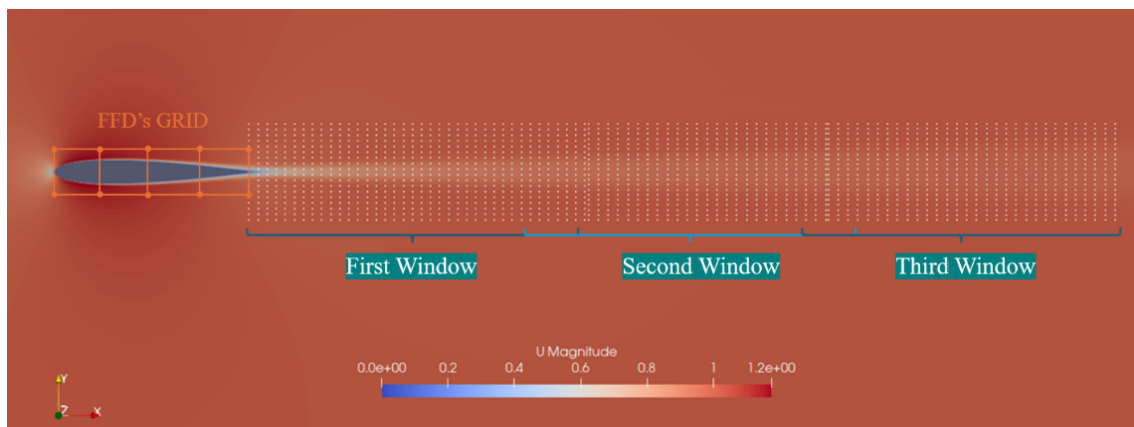


Figura 8.1: Concetto dello studio. La forma del NACA0012 è identificata a partire da una firma di scia campionata da una finestra di probes progressivamente traslata a valle; anche la geometria di partenza viene variata. La chiusura S–A è fissa per il problema di riferimento e per quello inverso.

8.2 Dati di riferimento e configurazione target

A differenza dello studio parallelo referenziato su LES, il campo di velocità di riferimento che guida l’ottimizzazione inversa è qui generato da una simulazione RANS diretta del NACA0012 target, con la chiusura di Spalart–Allmaras (la stessa poi impiegata nel ciclo aggiunto). La condizione target è un numero di Reynolds basato sulla corda $Re = 5 \times 10^4$ e un angolo di attacco $AoA = 0$, con velocità di flusso libero $U_\infty = 1$ m/s e viscosità cinematica $\nu = 2 \times 10^{-5}$ m²/s. L’angolo di attacco è imposto al contorno di far-field decomponendo la velocità di flusso libero in componenti cartesiane $U_x = U_\infty \cos \alpha$ e $U_y = U_\infty \sin \alpha$; a $\alpha = 0$ si ha $U_x = 1$, $U_y = 0$.

Poiché sia il campo di riferimento sia tutte le simulazioni inverse successive sono prodotti dallo stesso solutore incompressibile (DASimpleFoam) con la stessa chiusura e alle medesime condizioni, non vi è alcuna incongruenza di modello, di comprimibilità o di famiglia di solutore tra dati e modello inverso. La configurazione è quindi pienamente autoconsistente, e ogni discrepanza residua nella forma ricostruita è attribuibile alla configurazione di osservazione (posizione della finestra di probes) e alla geometria di partenza, le due variabili in studio.

8.2.1 Costruzione del campo di riferimento (UData)

La soluzione S–A convergente attorno al NACA0012 target fornisce il campo di velocità di riferimento. Per renderlo utilizzabile all’interno dell’obiettivo di varianza di DAfoam, il campo è mappato su ciascuna mesh di partenza tramite l’utilità OpenFOAM `mapFields`, invocata con il metodo `cellPointInterpolate` e i flag `consistent` e `noFunctionObjects`, prendendo come sorgente l’ultimo istante temporale disponibile della simulazione diretta. Il flag `consistent` indica che le mesh sorgente e destinazione condividono la stessa geometria di contorno, per cui non viene applicata alcuna trasformazione geometrica. Il risultato è scritto in una directory fittizia `0` del caso destinazione; da questo insieme si trattiene la sola velocità U , rinominata `UData`, il campo vettoriale atteso dall’obiettivo DAfoam. Le celle della mesh destinazione che cadono fuori dal supporto spaziale del campo sorgente sono assegnate automaticamente al valore di flusso libero.

Ogni ottimizzazione confronta poi la propria velocità simulata con lo stesso campo UData: l'unica quantità che varia lungo lo sweep è quali coordinate di probe cadono nella finestra di osservazione attiva.

8.3 Geometrie di partenza

Come geometrie iniziali dell'ottimizzazione inversa si selezionano tre profili NACA a quattro cifre, tutti finalizzati a recuperare il NACA0012 target: NACA0008 (più sottile, simmetrico), NACA0015 (più spesso, simmetrico) e NACA2412 (camber moderato). Questa selezione copre un intervallo di distanze geometriche dal target, da una deformazione puramente di spessore (0008, 0015) a una combinata di spessore e camber (2412), consentendo di valutare l'influenza della stima iniziale sul problema inverso.

Tutte le coordinate dei profili sono generate analiticamente dalle equazioni standard di spessore e linea media dei NACA a quattro cifre, con spaziatura a coseno per infittire i punti verso i bordi d'attacco e di fuga. Ogni profilo è troncato a $x/c = 0.998$ per produrre un bordo di fuga tronco (*blunt*), richiesto dallo strumento di estrusione iperbolica della mesh (pyHyp), e chiuso da un segmento breve e quasi verticale.

8.3.1 Generazione della mesh

Per ciascuna geometria di partenza si ottiene una mesh superficiale strutturata mediante riparametrizzazione con spline cubiche e infittimento geometrico verso il bordo d'attacco (rapporto di crescita 1.3) e di fuga (rapporto 1.1), con dimensione massima di cella in direzione streamwise di 5×10^{-3} corde. La curva è estrusa in una mesh volumetrica 3D di tipo O con pyHyp, usando una distanza di prima cella a parete di 7×10^{-4} corde, 60 punti in direzione normale alla parete con rapporto di crescita ≈ 1.12 e una distanza di marcia di 20 corde, che colloca il far-field a circa 20 corde dal profilo. A $Re = 5 \times 10^4$ questa spaziatura è wall-resolved senza funzioni di parete, con $y^+ \approx 1.9$ in media (massimo ≈ 4.0) sul NACA0012 di riferimento. La mesh è convertita in formato OpenFOAM e suddivisa con autoPatch in quattro patch: due piani di simmetria spanwise ($z = 0$, $z = 0.1$), la parete del profilo (wing) e il far-field (inout).

Tabella 8.1: Parametri di qualità delle tre mesh di partenza.

Profilo	Celle	y^+ medio	Aspect ratio max	Skewness max
NACA0008	26 603	0.18	990.5	3.21
NACA0015	26 603	0.21	740.1	2.18
NACA2412	26 603	0.22	950.3	2.60

8.3.2 Nota sul bordo di fuga tronco

Poiché la distribuzione di spessore dei NACA a quattro cifre ha curvatura nulla ma pendenza non nulla a $x/c = 1$, la chiusura del profilo troncato con un segmento rettilineo introduce uno spigolo (circa 80–85 tra la superficie del dorso e il segmento di chiusura), indipendentemente dal punto di troncamento. Si tratta di una caratteristica strutturale di qualsiasi parametrizzazione a bordo di fuga tronco, non di un artefatto di meshatura: autoPatch identifica la chiusura come una regione distinta, che viene poi unita alla parete principale in createPatchDict per ottenere un'unica patch di parete continua.

8.4 Metodo numerico

8.4.1 Equazioni di governo e solutore

Il flusso è governato dalle equazioni RANS stazionarie e incompressibili, chiuse con il modello a una equazione di Spalart–Allmaras [32]. Esse sono discretizzate con il metodo dei volumi finiti e risolte con l’algoritmo SIMPLE tramite il solutore differenziabile DASimpleFoam del framework DA Foam [20, 33], costruito su OpenFOAM. Tutte le simulazioni sono pseudo-2D, eseguite su una mesh con un unico strato di celle in direzione spanwise, delimitato da due piani di simmetria. La convergenza del primale è monitorata attraverso il residuo massimo tra le variabili di stato ($\mathbf{U}$, p e la variabile S–A) e l’iterazione SIMPLE è considerata convergente quando questo scende sotto 10^{-11} ; l’equazione della pressione, con un fattore di sotto-rilassamento conservativo (0.30), è la variabile a convergenza più lenta.

8.4.2 Analisi di sensitività aggiunta

Le sensitività di forma sono calcolate con il metodo dell’aggiunto discreto implementato in DA Foam, *identico* a quello derivato e discusso nel Capitolo 3: l’equazione aggiunta e la formula della sensitività totale non vengono qui ripetute. Anche in questo caso il sistema aggiunto è risolto con GMRES matrix-free, preconditionato con fattorizzazione ILU (fill level 1) e riordinamento RCM, e le derivate parziali sono valutate per differenziazione algoritmica in modalità inversa. L’unica specificità del presente caso è che l’operatore aggiunto $(\partial \mathbf{R} / \partial \mathbf{u})^T$ include ora la linearizzazione del modello di turbolenza S–A. Poiché un’unica chiusura fissa è usata sia per il riferimento sia per il problema inverso, l’operatore aggiunto appartiene alla stessa famiglia in tutto il processo e le sensitività sono consistenti per costruzione — in contrasto con il confronto multi-chiusura dello studio parallelo, dove chiusure diverse producevano campi aggiunti diversi sulla stessa soluzione primale.

8.4.3 Funzione obiettivo

La funzione obiettivo è la varianza tra il campo di velocità simulato $\mathbf{U}$ e il campo di riferimento $\mathbf{U}_{\text{data}}$, valutata nei probes contenuti nella finestra di osservazione attiva (Sezione 8.6). La sua forma è del tutto analoga a quella introdotta nel Capitolo 2 e non viene ridiscussa, con un’unica differenza sostanziale: in questo studio si adotta la formulazione *Variance classica* (non normalizzata) invece della *VarianceScaled* usata per il corpo tozzo. Esplicitamente,

$$J = \sum_i \left[(U_x(\mathbf{x}_i) - U_{x,\text{data}}(\mathbf{x}_i))^2 + (U_y(\mathbf{x}_i) - U_{y,\text{data}}(\mathbf{x}_i))^2 \right], \quad (8.1)$$

sommata sui punti di probe $\mathbf{x}_i$ e limitata alle componenti streamwise e trasversale. Non è aggiunto alcun termine di penalità su resistenza o portanza: il funzionale si basa esclusivamente sul campo di velocità, isolando l’effetto del posizionamento dei probes su un problema inverso puramente basato sul campo. A $\text{AoA} = 0$ la U_y di riferimento è simmetrica e piccola, per cui il deficit streamwise domina il segnale (si veda la Sezione 8.5).

8.4.4 Parametrizzazione della geometria (FFD)

La geometria è parametrizzata con un reticolo di Free-Form Deformation di 5 punti di controllo lungo la corda e 2 lungo lo spessore, immerso in un box leggermente più grande

del profilo ($x/c \in [-0.01, 1.01]$, $y/c \in [-0.09, 0.09]$) e replicato ai due piani di simmetria spanwise (Figura 8.2). Le tre colonne *interne* muovono liberamente i punti sul lato in pressione e in depressione in direzione y : uno spostamento di segno opposto dei due punti ispessisce o assottiglia localmente il profilo, indipendentemente dal fatto che la geometria di partenza debba ispessirsi o assottigliarsi per avvicinarsi al target. Le colonne di *bordo* (bordo d'attacco e di fuga) non sono invece completamente bloccate: i loro due punti sono liberi di traslare verticalmente, ma sono obbligati a farlo in modo *simmetrico* (spostamenti uguali e opposti), così da lasciare che la curvatura locale si modifichi senza però traslare verticalmente il punto di bordo d'attacco e di fuga. La deformazione della mesh è propagata al volume con lo strumento IDWarp (*inverse-distance weighting*), limitando le superfici mobili alla parete del profilo così che il far-field resti indeformato.

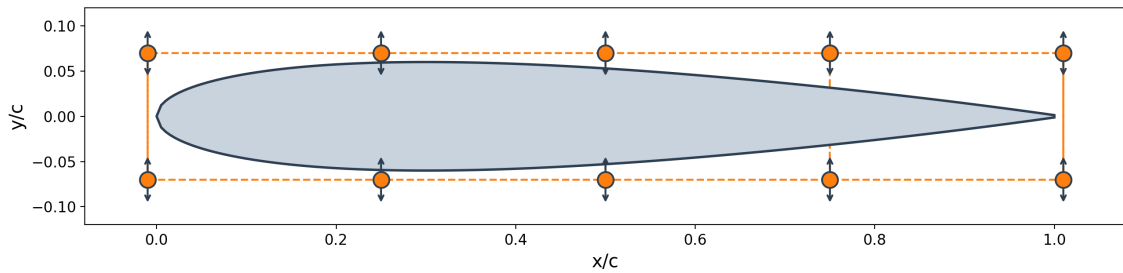


Figura 8.2: Reticolo FFD 5×2 attorno al profilo. Tutti i punti di controllo (cerchi arancioni, con frecce verticali) possono traslare in y . Le tre colonne interne si muovono liberamente e indipendentemente sui due lati (ispessimento o assottigliamento locale, con eventuale asimmetria); le colonne di bordo d'attacco e di fuga sono invece vincolate a muoversi in modo simmetrico (spostamenti uguali e opposti dei due punti), così da modificare la curvatura di bordo soprattutto al bordo d'attacco, senza traslare verticalmente i punti di bordo d'attacco e di fuga.

8.4.5 Vincoli geometrici

Due vincoli geometrici impediscono forme non fisiche, auto-intersecanti o eccessivamente distorte:

- un vincolo di volume relativo (area della sezione trasversale in 2D), valutato con DVConstraints di pyGeo su una superficie triangolata, limitato tra 0.85 e 1.5 volte il valore della geometria di partenza;
- un vincolo di spessore relativo puntuale in 10 stazioni lungo la corda (escludendo un piccolo margine ai bordi), limitato tra 0.5 e 2.0 volte il valore di partenza in ciascuna stazione.

Entrambi sono imposti come vincoli di disuguaglianza non lineari all'interno di IPOPT, insieme ai vincoli di box sulle variabili di progetto.

8.4.6 Algoritmo di ottimizzazione

Il problema è risolto con l'Interior Point OPTimizer (IPOPT) [29], con approssimazione L-BFGS dell'Hessiana, aggiornamento adattivo del parametro di barriera e globalizzazione a filtro con line-search. I gradienti dell'obiettivo e dei vincoli sono forniti a ogni

iterazione dalla procedura aggiunta discreta della Sezione 8.4.2. L'ottimizzazione è ripetuta indipendentemente per ciascuna geometria di partenza e ciascuna posizione della finestra di osservazione. Ogni run si arresta a un tetto fisso di iterazioni (usato come soglia uniforme per rendere confrontabili i run) oppure prima, tramite la regola di arresto basata sull'errore geometrico della Sezione 8.6.3.

8.5 Configurazione del flusso e della chiusura turbolenta

Una singola chiusura turbolenta è impiegata in tutto lo studio, il modello di Spalart–Allmaras, sia per il campo di riferimento del NACA0012 sia per l'ottimizzazione inversa delle tre geometrie di partenza. Il valore di far-field della viscosità turbolenta modificata $\tilde{\nu}$ è posto pari a tre volte la viscosità cinematica molecolare ν , scelta standard corrispondente a una bassa intensità di turbolenza del flusso libero. Il modello è eseguito senza funzioni di parete, con una distanza di prima cella progettata per $y^+ \approx 2$ (Sezione 8.3.1). I parametri complessivi del caso sono riassunti nella Tabella 8.2.

Tabella 8.2: Configurazione fisica e numerica dello studio sul profilo alare.

Target / partenza	Target: NACA0012 Partenza: NACA0008, NACA0015, NACA2412
Chiusura turbolenta	Spalart–Allmaras (fissa per riferimento + inverso)
Numero di Reynolds	$Re = 5 \times 10^4$ (basato sulla corda)
Flusso libero	$U_\infty = 1$ m/s, $\nu = 2 \times 10^{-5}$ m ² /s
Angolo di attacco	0 ($U_x = 1$, $U_y = 0$)
S–A far field	$\tilde{\nu} = 3\nu = 6 \times 10^{-5}$
Trattamento a parete	Low-Re, no wall functions, $y^+ \approx 1.9$ (max ≈ 4.0)
Solutore / accoppiamento	DASimpleFoam (stazionario, incomprimibile), SIMPLE, pseudo-2D

8.5.1 Scelta del numero di Reynolds e compromesso sulla mesh di parete

Il numero di Reynolds di lavoro, $Re = 5 \times 10^4$, è inferiore al valore adottato in alcuni studi di inverse design di riferimento ($Re = 1 \times 10^5$). La scelta è deliberata e discende da un compromesso tra il numero di Reynolds e la qualità di mesh ottenibile. A $Re = 1 \times 10^5$ lo strato limite è sottile, e la spaziatura a parete ereditata dalla configurazione a Reynolds più basso (prima cella $\approx 1 \times 10^{-3}$ corde) dava $y^+ \approx 5$ sul profilo, troppo grossolana per un calcolo Spalart–Allmaras wall-resolved a basso Reynolds senza funzioni di parete. Ripristinare $y^+ \approx 1$ richiedeva ridurre la distanza di prima cella di circa un fattore cinque; sul bordo di fuga tronco e nelle regioni ad alta curvatura ciò produceva celle con aspect ratio di diverse migliaia, oltre il limite di default di `checkMesh` e, più criticamente, celle quasi degeneri (volumi prossimi a zero) che lo step di deformazione della mesh (IDWarp) rifiutava, impedendo persino l'avvio dell'ottimizzazione. Mantenere invece la spaziatura di parete più grossolana lasciava y^+ troppo alto: i residui del primale ristagnavano attorno a 7×10^{-6} con la variabile S–A ripetutamente clippata, il che a sua volta impediva la convergenza del solutore aggiunto GMRES (che stallava attorno a 10^{-3}).

Abbassare il Reynolds a 5×10^4 rimuove questo conflitto. Lo strato limite è circa il 40% più spesso che a 1×10^5 , per cui una prima cella di 7×10^{-4} corde produce già una

mesh wall-resolved ($y^+ \approx 1.9$ in media, massimo ≈ 4.0) mantenendo le celle di bordo di fuga prossime all'aspect ratio unitario: `checkMesh` passa, `IDWarp` deforma la mesh senza fallire e il primale converge sotto 10^{-6} con la variabile S-A non più clippata. La ridotta rigidità del sistema a Reynolds più basso migliora anche il condizionamento dell'aggiunto, e GMRES converge. Fisicamente, $Re = 5 \times 10^4$ resta un regime pienamente appropriato: il modello S-A è ben all'interno del suo campo di calibrazione, il flusso a $AoA = 0$ è attaccato senza bolla di separazione, e la scia porta ancora un deficit di velocità forte e ben condizionato: esattamente il segnale su cui si basa lo sweep di posizionamento dei probes. Anzi, la scia leggermente più spessa a questo Reynolds è vantaggiosa per lo sweep a valle, poiché il deficit persiste più a lungo prima di decadere.

Flusso attaccato a $AoA = 0$. A incidenza nulla il NACA0012 simmetrico non genera portanza e sviluppa solo un debole gradiente di pressione avverso simmetrico. La bolla di separazione laminare, presente in flussi transizionali sul dorso a incidenza moderata, è un fenomeno guidato dall'incidenza e dalla transizione; inoltre il modello S-A è pienamente turbolento e non rappresenta le bolle di separazione laminare, e il suo strato limite turbolento resiste alla separazione. Di conseguenza, a $AoA = 0$ e $Re = 5 \times 10^4$ la soluzione S-A del NACA0012 resta attaccata, senza bolla di separazione visibile, e lo stesso vale per le geometrie di partenza deformate, il cui spessore/camber resta in un intervallo moderato. La firma di scia prossima è quindi sottile e simmetrica (Figura 8.3). Poiché S-A è usato in modo identico per il riferimento e per il modello inverso, questa rappresentazione attaccata e pienamente turbolenta è condivisa da entrambi: i dati sono autoconsistenti e non affetti da bias rispetto al modello inverso.

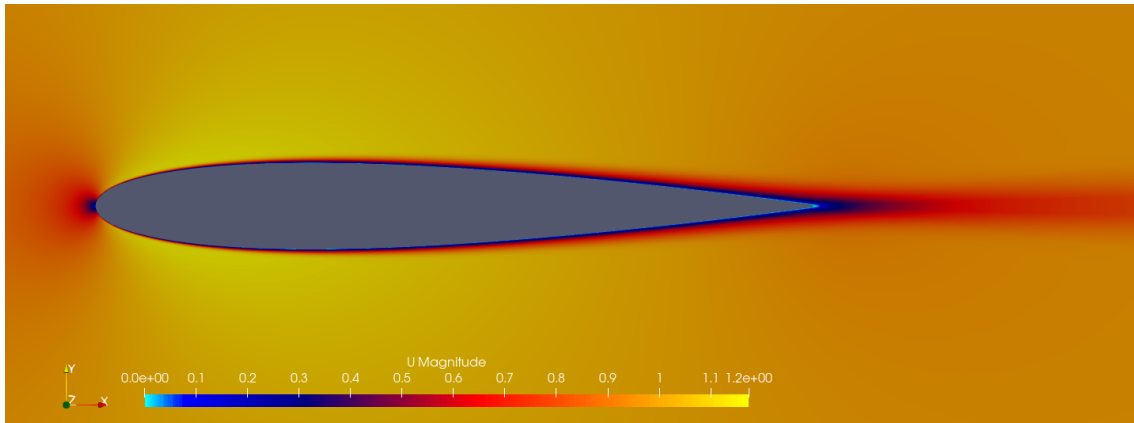


Figura 8.3: Verifica che la soluzione S-A del NACA0012 target a $AoA = 0$ e $Re = 5 \times 10^4$ sia attaccata, così che la firma di scia di riferimento sia pulita e simmetrica. L'unica regione a velocità prossima allo zero (con caratteristiche di piccola bolla di separazione) si trova in corrispondenza della porzione mancante del bordo di fuga tronco.

8.6 Finestra di osservazione e strategia di posizionamento dei probes

8.6.1 Finestra di base e traslazione a valle

Lo studio testa tre finestre di osservazione, ciascuna copre una regione rettangolare di $2 \times 0.5 c^2$ nella scia ($y/c \in [-0.25, 0.25]$), progressivamente traslata a valle: prima (W1)

$x/c \in [1.01, 3.0]$, seconda (W2) $x/c \in [2.5, 4.5]$, terza (W3) $x/c \in [4.0, 6.0]$ (Tabella 8.3). Il bordo anteriore della prima finestra si colloca a $x/c = 1.01$, immediatamente a valle del corpo (la corda termina a $x/c = 1$). Le tre finestre sono poste a posizioni a valle fisse, mantenendone invariati dimensione e numero di probes, e il problema inverso è risolto nuovamente per ciascuna posizione. L'obiettivo è trovare il maggiore offset a valle per cui la ricostruzione resta quasi identica al target. Fisicamente, il deficit di velocità streamwise e la sua struttura laterale decadono con la distanza, per cui il segnale di scia e quindi il condizionamento del problema inverso indeboliscono man mano che la finestra si allontana; la finestra di campo vicino, viceversa, si sovrappone alla regione del bordo di fuga dove il flusso è più sensibile, in eco alla sensibilità della bolla di ricircolazione osservata nel benchmark ellisse $\rightarrow$ cilindro.

Tabella 8.3: Le tre finestre di osservazione. Tutte condividono $y/c \in [-0.25, 0.25]$ e sono estratte da un file di coordinate comune che copre $x/c = 1.01-6.0$.

Finestra	intervallo x/c	intervallo y/c	Δx dal TE
Prima (W1)	[1.01, 3.0]	$[-0.25, 0.25]$	0.01 (al TE)
Seconda (W2)	[2.5, 4.5]	$[-0.25, 0.25]$	1.50
Terza (W3)	[4.0, 6.0]	$[-0.25, 0.25]$	3.00

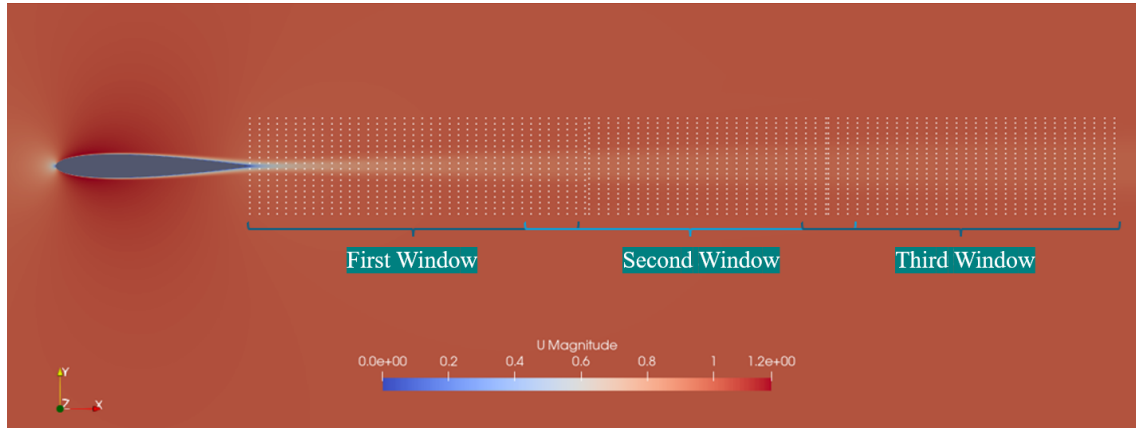


Figura 8.4: Layout rappresentativo dei tre test di finestra a valle, mostrato sulla scia del NACA0012 (usato qui come riferimento generico per tutte le geometrie di partenza): primo test $x/c \in [1.01, 3.0]$, secondo $x/c \in [2.5, 4.5]$, terzo $x/c \in [4.0, 6.0]$. Le stesse tre finestre sono applicate invariate a ogni profilo di partenza.

8.6.2 Generazione delle coordinate dei probes

Le coordinate dei probes all'interno di ciascuna finestra sono lette da un file predisposto in modo che nessun probe cada dentro un solido: per le finestre di campo vicino il fronte del box ($x/c = 1.01$; $y/c \in [-0.25, 0.25]$) giace perpendicolare alla corda del profilo. La griglia è di 40×20 punti (uniforme entro la finestra); lo stesso numero di probes è mantenuto per ogni finestra, così che la magnitudine dell'obiettivo sia confrontabile lungo lo sweep.

8.6.3 Metrica di errore geometrico e soglia di arresto

Due metriche complementari quantificano il risultato. La metrica di forma primaria è l'errore L_2 nella coordinata y del profilo ottimizzato rispetto al NACA0012 target, valutato su un insieme comune di stazioni x/c su entrambi i lati,

$$L_2 = \sqrt{\frac{1}{N} \sum_k (y_k - y_{k,\text{target}})^2}. \quad (8.2)$$

La funzione obiettivo di varianza J dell'equazione (8.1) è riportata a fianco di L_2 come punteggio diretto (basato sul flusso). Come diagnostica complementare puramente geometrica, ereditata dal benchmark ellisse $\rightarrow$ cilindro, la deviazione di superficie può anche essere espressa in coordinate polari e sintetizzata da RMSE, MAE ed errore massimo; in questo studio, tuttavia, ci si è limitati alla sola metrica L_2 . La coppia RMSE/max guida l'eventuale arresto anticipato: quando l'errore geometrico scende sotto una tolleranza prescelta per un dato numero di iterazioni, l'ottimizzazione è interrotta per non sprecare soluzioni primali/aggiunte. In questo studio la soglia non è stata usata, ma la sua implementazione è cruciale per l'influenza che ha su un problema di inverse design (cfr. Sezione 8.7.3).

8.7 Risultati

8.7.1 Panoramica

Le sezioni seguenti presentano i risultati di ricostruzione per ciascuna delle tre geometrie di partenza (NACA0008, NACA0015, NACA2412) attraverso le tre finestre di probes (W1 $\rightarrow$ W3). Le Tabelle 8.4–8.6 riassumono le metriche chiave; la Tabella 8.7 riporta le distanze geometriche iniziali dal target. La migliore ricostruzione singola dell'intero studio è ottenuta da NACA0015/W2 ($L_2 = 3.08 \times 10^{-3}$), seguita da vicino da NACA0008/W1 ($L_2 = 3.11 \times 10^{-3}$). La geometria NACA2412 produce errori uniformemente elevati in tutte le finestre ($L_2 \approx 13\text{--}15 \times 10^{-3}$), a causa dell'incapacità della scia simmetrica a 0 di vincolare il camber. La Sezione 8.7.4 confronta direttamente le tre geometrie; la Sezione 8.7.5 discute il comportamento di convergenza e il criterio di arresto.

8.7.2 Effetto della posizione streamwise della finestra

Geometria di partenza NACA0008. Fissata la geometria di partenza (NACA0008) e tutte le impostazioni numeriche, si trasla la sola finestra di osservazione a valle attraverso tre posizioni. La Figura 8.5 mostra i profili ricostruiti per ciascuna posizione; la Tabella 8.4 ne riassume i risultati. La finestra del primo test, che inizia a $x/c = 1.01$ (immediatamente a valle del bordo di fuga) e si estende a $x/c = 3$, fornisce la ricostruzione più accurata: $L_2 = 3.11 \times 10^{-3}$, con l'ottimizzatore che converge in 21 iterazioni e riduce l'obiettivo di varianza di quasi un ordine di grandezza. Il profilo ricostruito segue da vicino il NACA0012 target su tutta la corda. Le finestre del secondo e terzo test, poste più a valle ($x/c \in [2.5, 4.5]$ e $[4.0, 6.0]$), producono errori geometrici sostanzialmente più alti: $L_2 = 5.71 \times 10^{-3}$ e 5.56×10^{-3} . Il terzo test supera leggermente il secondo pur essendo più lontano dal profilo. Questo risultato controintuitivo riflette due effetti concomitanti: la varianza iniziale a quelle distanze è già piccola (9.1×10^{-5} e 4.4×10^{-5}), il che significa che le firme di scia di NACA0008 e NACA0012 sono quasi indistinguibili a $x/c > 3$, per cui

il paesaggio dell'obiettivo è piatto e l'ottimizzatore converge a minimi locali leggermente diversi; e la terza finestra copre un intervallo in cui la firma di resistenza integrata fornisce un segnale di gradiente marginalmente più strutturato. Il risultato chiave è che oltre $x/c \approx 3$ il deficit di scia è decaduto a sufficienza che l'errore di ricostruzione satura su un plateau di circa $L_2 \approx 5.6 \times 10^{-3}$, indipendente dalla posizione esatta della finestra. Questo definisce il limite pratico a valle per un recupero di forma affidabile con questa configurazione di probes.

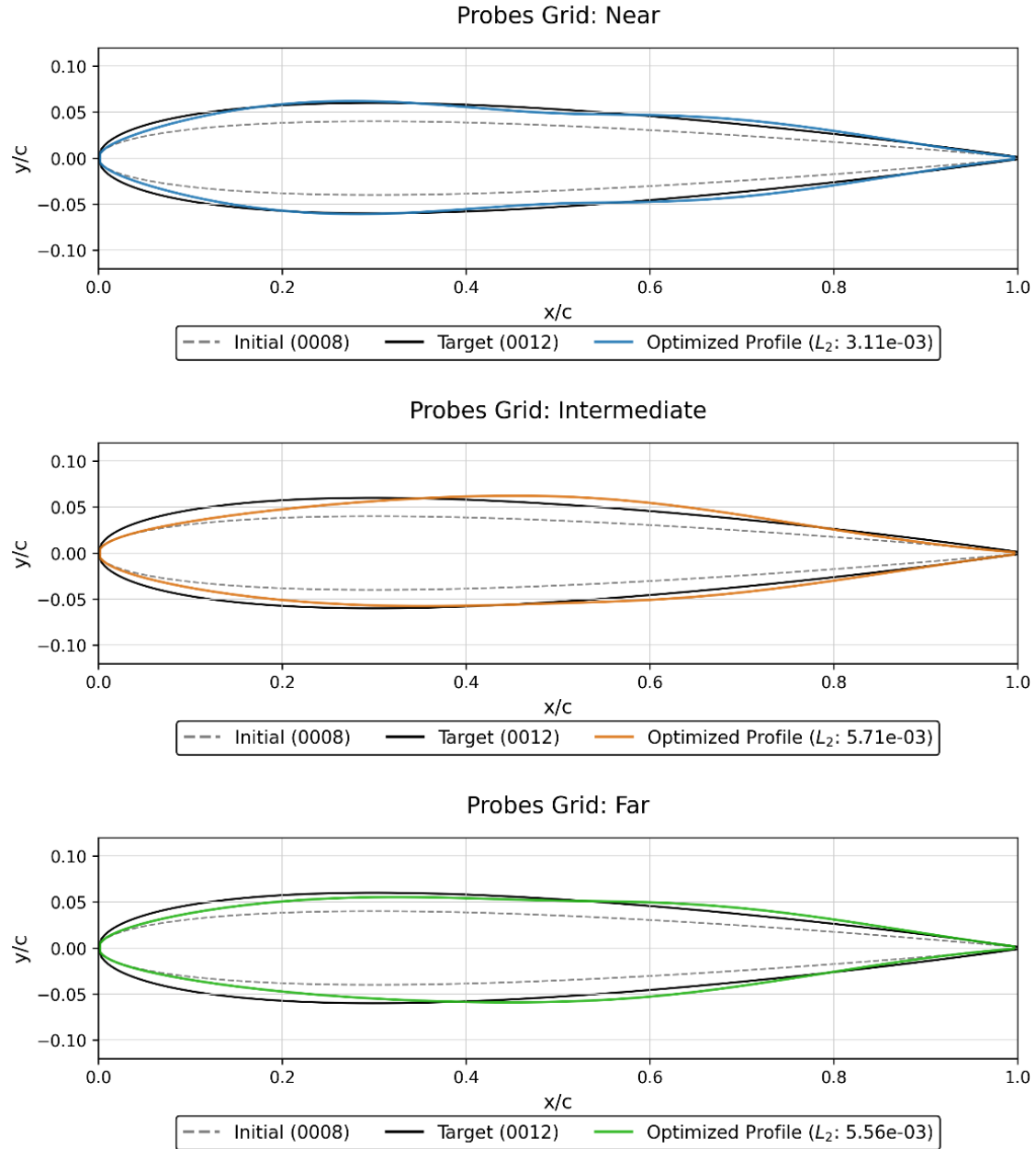


Figura 8.5: Profili ricostruiti per le tre finestre a valle (profilo di partenza NACA0008). In alto: primo test [1.01, 3.0], $L_2 = 3.11 \times 10^{-3}$; al centro: secondo [2.5, 4.5], $L_2 = 5.71 \times 10^{-3}$; in basso: terzo [4.0, 6.0], $L_2 = 5.56 \times 10^{-3}$. Il NACA0012 target è in nero.

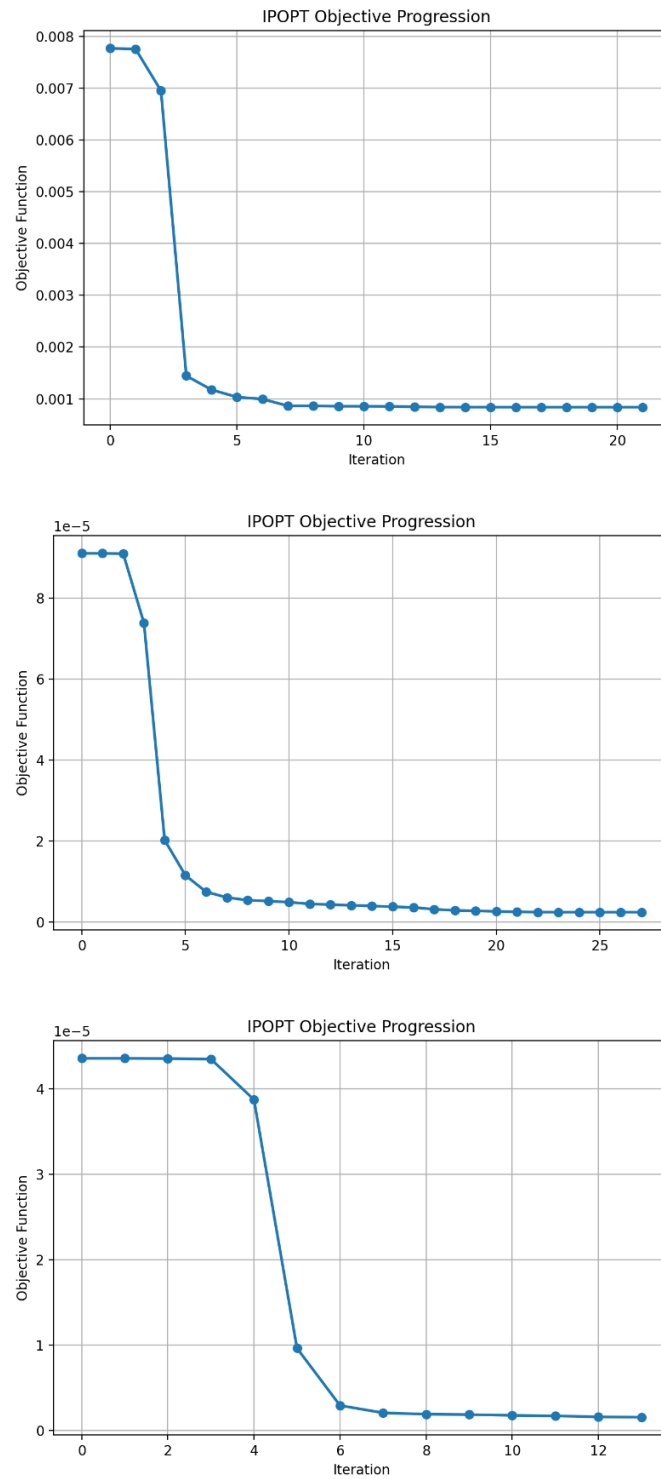


Figura 8.6: Progressione della funzione obiettivo IPOPT per i tre test di finestra (partenza NACA0008). In alto a sinistra: primo test $[1.01, 3.0]$, 21 iterazioni; in alto a destra: secondo test $[2.5, 4.5]$, 27 iterazioni; in basso: terzo test $[4.0, 6.0]$, 14 iterazioni. Si notino le diverse scale dell'asse verticale.

Geometria di partenza NACA0015. La Figura 8.7 mostra i profili ricostruiti per NACA0015 attraverso le tre finestre, e la Figura 8.8 le corrispondenti progressioni dell'obiettivo. Il comportamento di NACA0015 è opposto a quello di NACA0008: la finestra più vicina (W1) fornisce l'errore geometrico più alto ($L_2 = 6.32 \times 10^{-3}$), mentre la finestra intermedia (W2) dà il recupero migliore ($L_2 = 3.08 \times 10^{-3}$), con W3 che produce un risultato quasi identico ($L_2 = 3.11 \times 10^{-3}$). L'inversione ha una base fisica: NACA0015 è più spesso del target (0012) e deve assottigliarsi durante l'ottimizzazione, mentre NACA0008 deve ispessirsi. I probes di campo vicino a W1 campionano la regione del gradiente di deficit di spessore più forte; per la direzione di assottigliamento questo gradiente è mal condizionato vicino al bordo di fuga, guidando l'ottimizzatore in un minimo locale peggiore. Spostarsi a W2 leviga il campo di gradiente e produce una discesa meglio condizionata. È notevole che il miglior errore geometrico di NACA0015/W2 (3.08×10^{-3}) sia leggermente inferiore al migliore di NACA0008/W1 (3.11×10^{-3}), suggerendo che, per la finestra corretta, un profilo di partenza più spesso può convergere marginalmente più vicino al target.

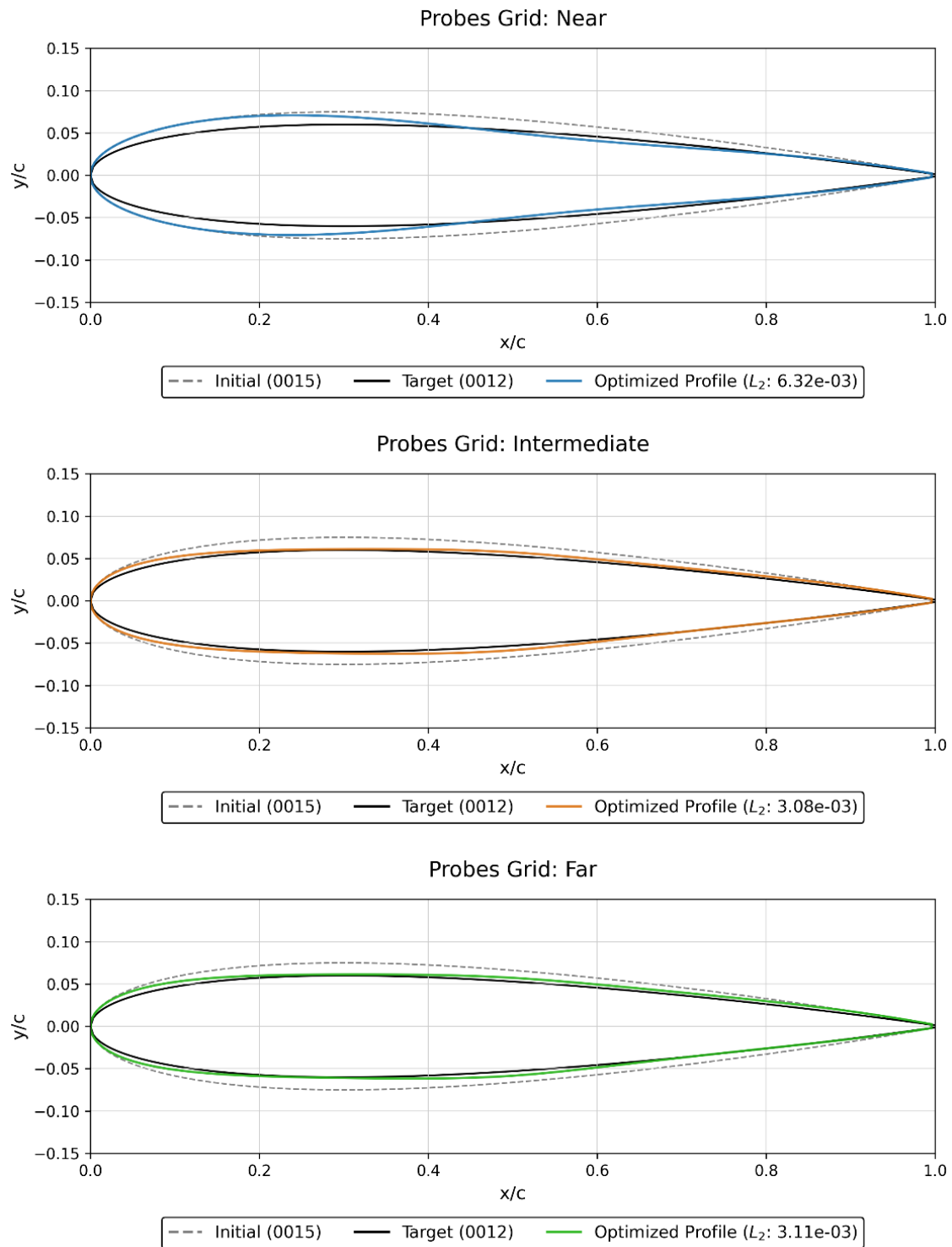


Figura 8.7: Geometria di partenza NACA0015: profili ricostruiti per le tre finestre. In alto: $W1$ (vicina) $L_2 = 6.32 \times 10^{-3}$; al centro: $W2$ (intermedia) $L_2 = 3.08 \times 10^{-3}$; in basso: $W3$ (lontana) $L_2 = 3.11 \times 10^{-3}$. La finestra vicina è la peggiore — l'opposto di NACA0008.

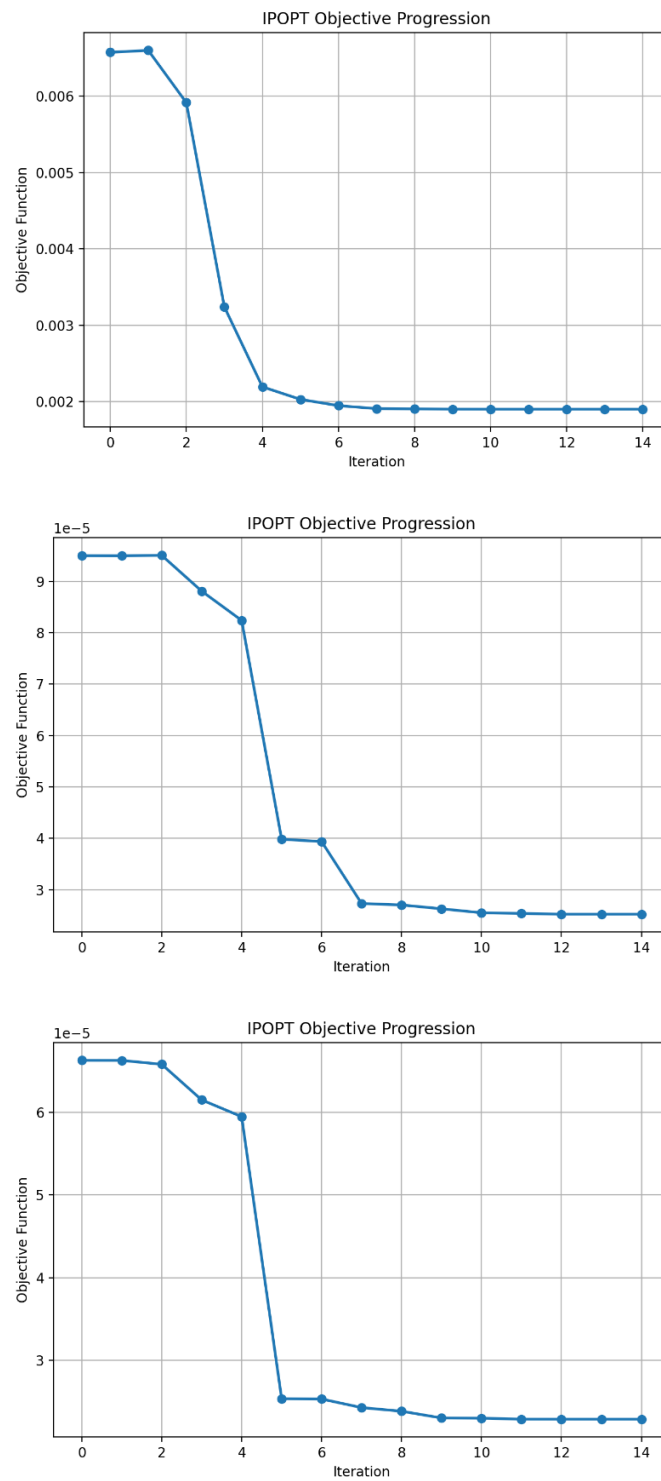


Figura 8.8: Progressioni dell'obiettivo IPOPT per la geometria NACA0015. In alto a sinistra: W1 (16 iter); in alto a destra: W2 (15 iter); in basso: W3 (15 iter).

8.7.3 NACA0015/W1: quasi-ottimo transitorio e divergenza geometrica

La Figura 8.9 mostra il profilo ricostruito per il caso NACA0015/W1 alla sesta iterazione ($L_2 = 3.73 \times 10^{-3}$). Sebbene questo risultato sia migliore di quello finale riportato per la stessa finestra ($L_2 = 6.32 \times 10^{-3}$), l'osservazione chiave è che attorno all'iterazione 6 il profilo era visivamente quasi coincidente con il NACA0012 target, prima che l'ottimizzatore lo riportasse verso una forma più spessa. Questo non è né un'instabilità numerica né un minimo locale: è una conseguenza diretta della disconnessione tra ciò che IPOPT minimizza (la varianza J del campo di velocità) e ciò che si misura (l'errore geometrico L_2). All'iterazione 6, J sta ancora diminuendo e le condizioni KKT non sono ancora soddisfatte, per cui IPOPT prosegue. Muove poi le variabili di progetto in una direzione che riduce ulteriormente J ma viaggia tangenzialmente al minimo di L_2 , peggiorando l'errore geometrico. La causa profonda è la degenerazione parziale della scia simmetrica a 0° : J è quasi piatta vicino all'ottimo geometrico, per cui molte forme diverse danno valori dell'obiettivo simili e l'ottimizzatore vaga in quella regione piatta. La traiettoria di ottimizzazione transita per la geometria vera $\mathbf{x}_{\text{true}}$ nel percorso verso il minimizzatore dell'obiettivo $\mathbf{x}_J^*$; ma in $\mathbf{x}_{\text{true}}$ il gradiente dell'obiettivo non è nullo, per cui l'ottimizzatore prosegue e la geometria si allontana. Il degrado non è dunque un'anomalia numerica, bensì la manifestazione diretta della non coincidenza tra “forma ottima di scia” e “forma vera” nelle regioni in cui la corrispondenza biunivoca è imperfetta: lo stesso meccanismo che, nel corpo tozzo, si osserva come disaccoppiamento tra obiettivo e accuratezza geometrica. Un criterio di arresto anticipato basato sulla geometria avrebbe catturato la forma corretta.

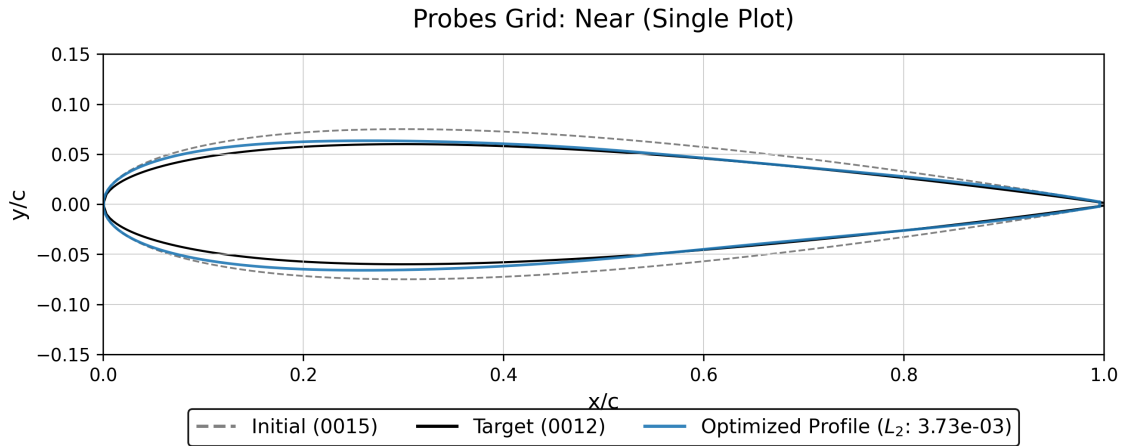


Figura 8.9: NACA0015/W1 (finestra vicina): profilo ricostruito finale (blu, $L_2 = 3.73 \times 10^{-3}$). Nonostante il discreto risultato finale, l'ottimizzatore è passato attraverso una corrispondenza geometrica quasi perfetta con NACA0012 attorno alle iterazioni 5–6, per poi allontanarsene perché il criterio di convergenza NLP di IPOPT ($\text{tol} = 10^{-6}$) non era ancora soddisfatto. Ciò illustra la disconnessione tra minimizzare J (varianza, obiettivo di IPOPT) e minimizzare L_2 (errore geometrico, la quantità di interesse).

Nota sul conteggio delle iterazioni. Il numero di iterazioni (15–20) è controllato esclusivamente dal criterio di convergenza NLP di IPOPT ($\text{Tol} = 10^{-6}$), non dalle tolleranze del primale o dell'aggiunto. Le impostazioni $\text{primalMinResTol} = 10^{-12}$ e $\text{gmresRelTol} = 10^{-10}$ governano solo l'accuratezza interna di ciascuna valutazione di funzione (soluzione CFD) e di gradiente (soluzione aggiunta): tolleranze interne più strette producono valutazioni più accurate, ma non cambiano il criterio di arresto. Il conteggio variabile di iterazioni riflette quanto rapidamente l'errore NLP complessivo di IPOPT raggiunge 10^{-6} , che dipende dalla curvatura dell'obiettivo, dalla distanza del punto di partenza dall'ottimo NLP e dallo schedule adattivo di μ . Casi con obiettivo quasi piatto (segnale di scia debole, 0) soddisfano le condizioni KKT con meno passi. Per contrasto, il caso ellisse $\rightarrow$ cilindro a $Re = 40$ superava le 100 iterazioni perché la trasformazione geometrica era ampia, i gradienti forti in tutto il processo e l'NLP genuinamente più difficile da convergere.

Geometria di partenza NACA2412. Le Figure 8.10 e 8.11 mostrano i risultati per NACA2412. Tutte e tre le finestre producono errori geometrici elevati: $L_2 = 1.32 \times 10^{-2}$ (W1), 1.50×10^{-2} (W2) e 1.50×10^{-2} (W3), sostanzialmente più alti di entrambe le geometrie simmetriche. La degradazione è primariamente attribuibile al camber del 2%, ma per ragioni legate all'ottimizzazione più che all'osservabilità. Un NACA2412 a $\text{AoA} = 0$ genera portanza, e quindi una scia asimmetrica con downwash e componente trasversale $U_y \neq 0$: rispetto al target NACA0012, a scia simmetrica, la discrepanza iniziale ai probes è dunque grande e il gradiente aggiunto è forte. Il camber è del resto pienamente osservabile, poiché una scia simmetrica corrisponde univocamente a portanza nulla e, a incidenza nulla, a camber nullo. Il recupero incompleto non deriva quindi da un'assenza di segnale, bensì dall'arresto dell'ottimizzatore in un minimo locale: la parametrizzazione FFD, con le sole tre colonne interne libere, fatica a rimuovere il camber (traslazione della linea media) senza accoppiarlo a variazioni di spessore, e IPOPT converge a un punto KKT con camber residuo. Le forme recuperate collassano così verso la componente simmetrica del target ma ne conservano una significativa asimmetria residua. Che il limite sia intrinseco alla forma e alla parametrizzazione, e non alla posizione dei sensori, è confermato dal fatto che tutte e tre le finestre danno $L_2 \approx 1.3\text{--}1.5 \times 10^{-2}$: se il fattore limitante fosse il segnale di scia, il risultato dipenderebbe fortemente dalla finestra. Vi contribuisce infine un effetto di condizionamento: essendo il target simmetrico, l'obiettivo è una funzione pari del camber, per cui il gradiente di rimozione, forte lontano dal target, si annulla man mano che il camber tende a zero, rendendo lenta l'eliminazione dell'ultima frazione di curvatura.

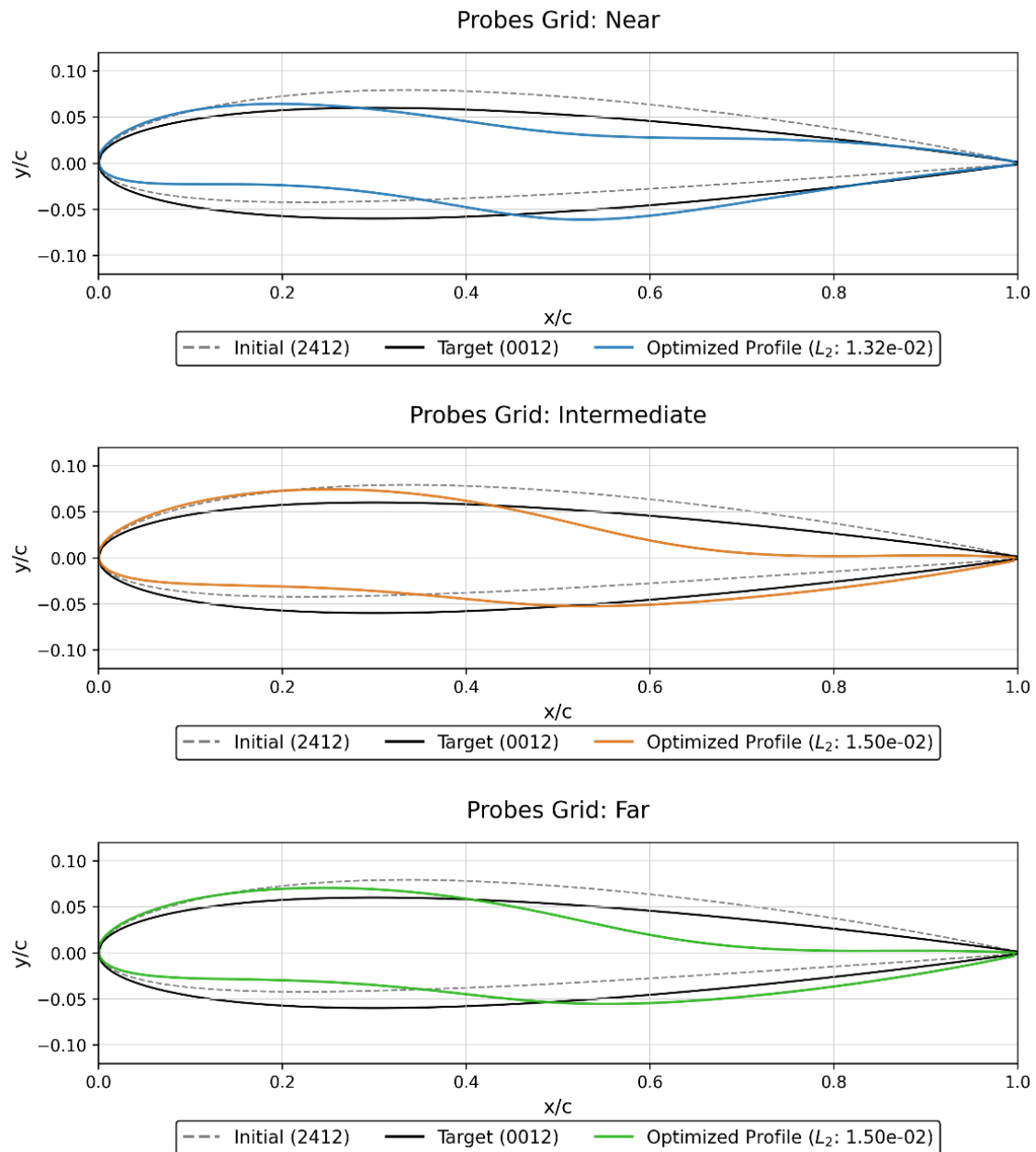


Figura 8.10: Geometria di partenza NACA2412: profili ricostruiti per le tre finestre. In alto: $W1$ (vicina) $L_2 = 1.32 \times 10^{-2}$; al centro: $W2$ (intermedia) $L_2 = 1.50 \times 10^{-2}$; in basso: $W3$ (lontana) $L_2 = 1.50 \times 10^{-2}$. Tutti gli errori sono un ordine di grandezza più grandi che per le geometrie simmetriche.

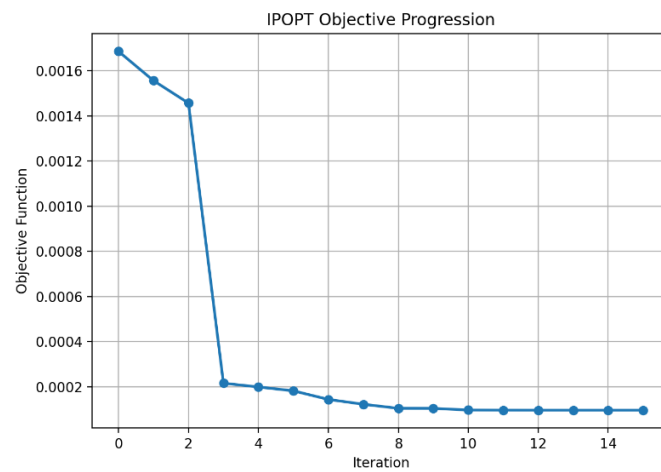
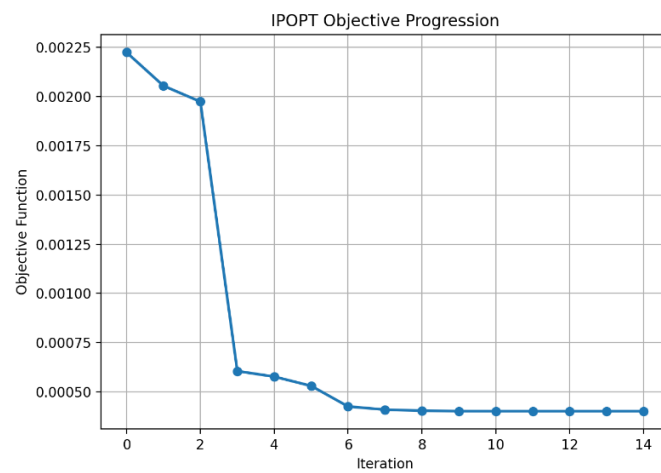
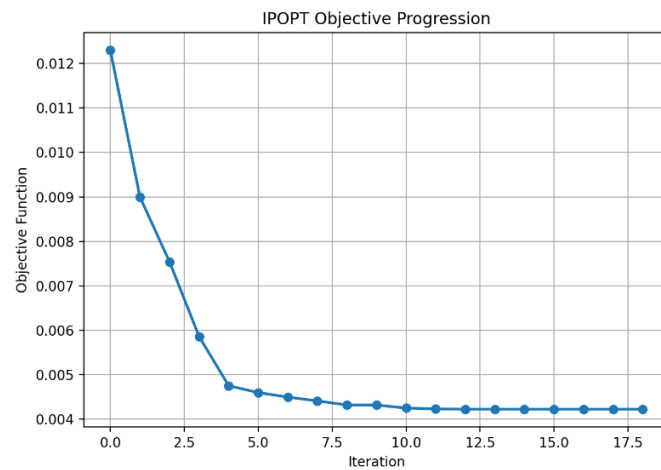


Figura 8.11: Progressioni dell'obiettivo IPOPT per la geometria NACA2412. In alto a sinistra: W1 (18 iter); in alto a destra: W2 (15 iter); in basso: W3 (16 iter).

Tabella 8.4: Finestra W1 ($x/c \in [1.01, 3.0]$, $y/c \in [-0.25, 0.25]$). J_0 : varianza iniziale; J^* : varianza finale; L_2 : errore geometrico rispetto al NACA0012.

Partenza	Iter	J_0	J^*	L_2 (y)
NACA0008	21	7.77×10^{-3}	8.39×10^{-4}	3.11×10^{-3}
NACA0015	16	6.57×10^{-3}	1.90×10^{-3}	6.32×10^{-3}
NACA2412	18	1.22×10^{-2}	4.23×10^{-3}	1.32×10^{-2}

Tabella 8.5: Finestra W2 ($x/c \in [2.5, 4.5]$, $y/c \in [-0.25, 0.25]$).

Partenza	Iter	J_0	J^*	L_2 (y)
NACA0008	27	9.10×10^{-5}	2.49×10^{-6}	5.71×10^{-3}
NACA0015	15	9.51×10^{-5}	2.50×10^{-5}	3.08×10^{-3}
NACA2412	15	2.22×10^{-3}	4.03×10^{-4}	1.50×10^{-2}

Tabella 8.6: Finestra W3 ($x/c \in [4.0, 6.0]$, $y/c \in [-0.25, 0.25]$).

Partenza	Iter	J_0	J^*	L_2 (y)
NACA0008	14	4.36×10^{-5}	1.55×10^{-6}	5.56×10^{-3}
NACA0015	15	6.61×10^{-5}	2.48×10^{-5}	3.11×10^{-3}
NACA2412	16	1.66×10^{-3}	9.60×10^{-5}	1.50×10^{-2}

8.7.4 Effetto della geometria di partenza

La Tabella 8.7 riassume le distanze geometriche iniziali. La finestra ottimale differisce tra le geometrie: W1 è la migliore per NACA0008 ($L_2 = 3.11 \times 10^{-3}$), W2 per NACA0015 ($L_2 = 3.08 \times 10^{-3}$). Questa dipendenza dalla direzione della finestra ottimale (ispessimento vs assottigliamento verso il target) è un risultato chiave: il contenuto informativo di una finestra di scia dipende non solo dalla sua posizione a valle ma anche dalla direzione di deformazione richiesta. NACA2412 non recupera il target in alcuna finestra ($L_2 \approx 13\text{--}15 \times 10^{-3}$) a causa del problema di osservabilità del camber.

Tabella 8.7: Distanza geometrica iniziale di ciascuna geometria di partenza dal NACA0012 target.

Partenza	errore L_2 (y), iniziale
NACA0008	1.89×10^{-2}
NACA0015	1.42×10^{-2}
NACA2412	1.35×10^{-2}

8.7.5 Comportamento di convergenza

Le progressioni dell'obiettivo IPOPT (Figure 8.6, 8.8, 8.11) confermano l'interpretazione basata sul contenuto informativo: il primo test converge in modo regolare su 21 iterazioni riducendo la varianza di quasi un ordine di grandezza, mentre il secondo e il terzo test partono da mismatch iniziali molto più piccoli (9×10^{-5} e 4×10^{-5}) e convergono in 27 e

14 iterazioni verso minimi locali poco profondi. In tutti e tre i casi IPOPT riporta EXIT: Optimal Solution Found con violazione dei vincoli nulla, a conferma che le forme recuperate soddisfano tutti i vincoli geometrici. L'estensione dello studio a un piccolo angolo di attacco positivo (2–4) è lasciata a sviluppi futuri: introdurre un'asimmetria di quantità di moto trasversale nella scia arricchirebbe il segnale di osservazione e potrebbe estendere la finestra utile oltre il $x/c \approx 3$ qui individuato.

8.8 Discussione

Poiché la chiusura è fissa e consistente, l'errore residuo di ricostruzione qui misurato riflette il contenuto informativo, il posizionamento dei probes e la geometria di partenza, piuttosto che una discrepanza di modello. Questa è la base pulita contro cui leggere le penalità di inconsistenza di chiusura dello studio parallelo referenziato su LES: là, un errore di un ordine di grandezza era guidato dal disallineamento tra la fisica generatrice dei dati (LES, con bolla di separazione laminare) e le chiusure RANS pienamente turbolente; qui quel disallineamento è rimosso per costruzione e l'errore rimanente è informativo.

Attraverso le tre geometrie di partenza lo studio rivela due regimi distinti. Per le geometrie simmetriche (0008 e 0015) il recupero di forma è possibile ma fortemente dipendente dalla finestra, con la posizione ottimale determinata dalla direzione di deformazione richiesta più che dalla semplice distanza: i probes di campo vicino (W1) favoriscono la direzione di ispessimento (0008), mentre i probes intermedi (W2) favoriscono la direzione di assottigliamento (0015). Oltre le rispettive finestre migliori, entrambe le geometrie simmetriche raggiungono un plateau $L_2 \approx 5\text{--}6 \times 10^{-3}$, un limite fissato dal contenuto informativo della scia simmetrica a 0. Per la geometria con camber (2412) il recupero è invece limitato non dall'osservabilità ma dall'ottimizzazione: la portanza generata a incidenza nulla produce una scia asimmetrica e quindi un segnale forte, ma la parametrizzazione FFD fatica a disaccoppiare camber e spessore e l'ottimizzatore si arresta in un minimo locale con camber residuo; coerentemente, tutte le finestre danno $L_2 \approx 1.3\text{--}1.5 \times 10^{-2}$ indipendentemente dalla posizione, segno che il limite è intrinseco alla forma e alla parametrizzazione e non alla misura. Un'anomalia di convergenza notevole è stata osservata per NACA0015/W1: l'ottimizzatore ha raggiunto una forma geometrica quasi ottimale all'iterazione 6, per poi deteriorarsi prima che il criterio di arresto NLP fosse soddisfatto. Ciò rimane coerente con la disconnessione tra le condizioni di ottimalità KKT di IPOPT e la metrica geometrica L_2 .

Limitazioni. Singolo numero di Reynolds; singolo target; RANS stazionario con chiusura pienamente turbolenta; osservazioni sintetiche prive di rumore; e, a $\text{AoA} = 0$, il segnale di scia più debole (simmetrico, portanza nulla). L'aggiunta di rumore di misura, di angoli di attacco multipli o di numeri di Reynolds multipli metterebbe ulteriormente alla prova la robustezza.

8.9 Conclusioni

Questo capitolo ha applicato il framework di identificazione inversa basato sull'aggiunto a una chiusura Spalart–Allmaras consistente, testando tre geometrie di partenza (NACA0008, NACA0015, NACA2412) attraverso tre finestre di probes a valle (W1 [1.01, 3.0], W2 [2.5, 4.5], W3 [4.0, 6.0]) a $Re = 5 \times 10^4$ e $\text{AoA} = 0$. Emergono tre conclusioni principali: (I) la finestra di probes ottimale è *dipendente dalla direzione* di deformazione:

W1 per le geometrie che devono ispessirsi, W2 per quelle che devono assottigliarsi; (II) entrambe le geometrie simmetriche raggiungono limiti di errore simili ($\approx 5-6 \times 10^{-3}$) nelle finestre non ottimali, rivelando un limite informativo comune della scia simmetrica a 0; (III) il recupero del camber (NACA2412) resta incompleto non per un limite di osservabilità, dato che il camber è pienamente determinato dalla scia asimmetrica generata dalla portanza, ma per l'arresto dell'ottimizzatore in un minimo locale, legato alla difficoltà della FFD di disaccoppiare camber e spessore. L'arresto anticipato basato sulla geometria è raccomandato per evitare la disconnessione KKT vs L_2 osservata per NACA0015/W1.

8.9.1 Posizionamento raccomandato dei probes e geometria migliore

- La finestra ottimale dipende dalla geometria di partenza: W1 [1.01, 3.0] per NACA0008 ($L_2 = 3.11 \times 10^{-3}$); W2 [2.5, 4.5] per NACA0015 ($L_2 = 3.08 \times 10^{-3}$).
- Il recupero del camber (NACA2412) resta incompleto in tutte le finestre ($L_2 \approx 13-15 \times 10^{-3}$): il limite non è di osservabilità ma di ottimizzazione (arresto in minimo locale, FFD che accoppia camber e spessore), e potrebbe essere mitigato arricchendo la parametrizzazione o migliorando il condizionamento (es. incidenza non nulla).
- Si raccomanda l'arresto anticipato geometrico (o la riduzione della tolleranza IPOPT): NACA0015/W1 mostra che l'ottimizzatore diverge geometricamente dopo aver raggiunto una forma quasi ottimale alle iterazioni 5-6, e si arresta prima di potervi eventualmente riconvergere.

Capitolo 9

Conclusioni e Sviluppi Futuri

Questa tesi ha sviluppato, validato e caratterizzato in dettaglio un metodo di *inverse design* aerodinamico basato sull'aggiunto discreto, con l'obiettivo di ricostruire la forma di un corpo a partire dalla sua firma di velocità nella scia. Il filo conduttore dell'intero lavoro è stato lo studio sistematico di come la qualità della ricostruzione dipenda dalle condizioni di osservazione [posizione, numero e densità dei probes] e dalla geometria di partenza, in un quadro reso rigoroso dall'adozione consapevole dell'*inverse crime*: il campo di riferimento e il ciclo di ottimizzazione condividono lo stesso risolutore, la stessa mesh e la stessa fisica, così da isolare l'effetto delle sole variabili di interesse ed eliminare qualunque bias di modello. Il metodo è stato applicato a due problemi di complessità crescente: la deformazione di un corpo tozzo bidimensionale (ellisse verso cilindro, laminare a $Re = 40$) e la ricostruzione di un profilo alare NACA in regime turbolento. Se ne raccolgono qui i risultati principali, le limitazioni e le prospettive di sviluppo.

9.1 Risultati sul corpo tozzo

L'analisi parametrica sul benchmark ellisse $\rightarrow$ cilindro (Capitoli 6–7) ha prodotto un quadro coerente e alcune conclusioni di validità generale.

In primo luogo, la regione utile per il posizionamento dei probes è delimitata *inferiormente* da un confine fisico non convenzionale coincidente con l'estensione della bolla di ricircolazione ($x \approx 3R$). Collocare sensori entro tale regione produce ricostruzioni gravemente distorte ($RMSE > 0.09$) indipendentemente dal numero di sensori, per una fragilità che si è rivelata di natura *strutturale*: la bolla è la zona di massima sensibilità del flusso quasi critico (prossimo alla soglia di distacco vorticoso del cilindro) e l'inversione di segno del campo di riferimento vi rende non convesso il problema; il termine di normalizzazione dell'errore relativo, che diverge in prossimità degli annullamenti del campo, agisce solo come amplificatore aggiuntivo per la formulazione normalizzata. *Superiormente*, invece, non esiste un limite pratico entro il dominio ideale: anche probes collocati a oltre venti raggi dal corpo ricostruiscono la forma con un errore di circa il 2.5% del raggio.

In secondo luogo, e più in profondità, la ricostruzione è risultata possibile soltanto laddove la mappa che lega la forma alla firma di velocità nella scia è *iniettiva e ben condizionata* sulla finestra di osservazione, ossia dove esiste un'effettiva corrispondenza biunivoca tra scia misurata e geometria. Ciò che spesso si descrive come “disaccoppiamento” tra valore finale della funzione obiettivo e accuratezza geometrica non è un fatto a sé, ma il *sintomo osservabile* della perdita di tale corrispondenza: dove la mappa si appiattisce lungo alcune direzioni dello spazio di forma (prossime al nucleo dell'operatore), forme geometricamente diverse producono scie quasi indistinguibili ai probes, cosicché l'obiettivo scende a valori piccolissimi mentre la geometria resta indeterminata. L'intera analisi parametrica va quindi letta come una *mappatura della regione di osservabilità*: l'intervallo di posizione, numero di probes e numero di Reynolds entro cui i cinque gradi di libertà della forma sono identificabili in modo ben condizionato. Che tale corrispondenza, quando esiste, sia largamente sovrabbondante è confermato dal successo della singola colonna di sensori; che venga meno per eccesso di distanza o all'interno della bolla è ciò che ne fissa i confini superiore e inferiore. È proprio questa natura del problema a rendere indispensabile una

metrica geometrica indipendente (l'RMSE radiale), poiché il solo valore dell'obiettivo non distingue una ricostruzione riuscita da una degenera: su di essa si fonda l'intera valutazione comparativa. La ricostruzione migliore in assoluto è stata ottenuta nella scia lontana a griglia piena (RMSE = 0.0054), che si è rivelata al tempo stesso la più accurata e la più rapida a convergere.

In terzo luogo, il confronto tra formulazioni della funzione obiettivo ha stabilito che la *VarianceScaled*, con normalizzazione locale, è superiore alla *Variance* classica in tutte le posizioni testate e, soprattutto, *robusta*: mantiene l'RMSE basso in ogni finestra, mentre la formulazione classica degrada sensibilmente quando la finestra include zone a bassa velocità. L'analisi del campo aggiunto ha infine fornito la chiave fisica di questi comportamenti: la qualità della ricostruzione è governata dalla *regolarità* della sensibilità, non dalla sua intensità. La scia prossima, pur essendo la regione a più alto contenuto informativo, genera un campo aggiunto intenso e concentrato attorno a pochi probes patologici, all'origine della distorsione; la scia lontana genera un campo aggiunto debole ma regolare, che guida una ricostruzione pulita. Il metodo si è inoltre dimostrato capace di operare persino con una singola colonna di sensori, a conferma che il problema è ampiamente sovradeterminato dal contenuto informativo di una sola sezione di scia.

9.2 Risultati sul profilo alare

L'estensione al profilo alare (Capitolo 8), con chiusura di turbolenza Spalart–Allmaras mantenuta fissa e consistente, ha confermato la trasferibilità della metodologia e ha rivelato fenomeni specifici del caso attaccato e turbolento.

Il risultato più significativo è che la finestra di osservazione ottimale è *dipendente dalla direzione* di deformazione richiesta, non dalla semplice distanza: i probes di campo vicino favoriscono la ricostruzione dei profili che devono ispessirsi (NACA0008), mentre i probes intermedi favoriscono quelli che devono assottigliarsi (NACA0015). Oltre la rispettiva finestra ottimale, entrambe le geometrie simmetriche raggiungono un plateau di errore comune ($L_2 \approx 5\text{--}6 \times 10^{-3}$), fissato dal contenuto informativo della scia simmetrica a incidenza nulla. Il caso con camber (NACA2412) resta invece la ricostruzione meno accurata in tutte le finestre ($L_2 \approx 1.3\text{--}1.5 \times 10^{-2}$); il limite, tuttavia, non è di osservabilità, poiché la portanza generata a incidenza nulla rende il camber pienamente leggibile nella scia asimmetrica, ma di ottimizzazione: la parametrizzazione FFD fatica a disaccoppiare camber e spessore e l'ottimizzatore si arresta in un minimo locale con curvatura media residua. Si è infine osservata, per il caso NACA0015/W1, una netta *disconnessione tra le condizioni di ottimalità KKT e la metrica geometrica*: l'ottimizzatore attraversa una forma quasi perfetta a metà del processo per poi allontanarsene, minimizzando l'obiettivo lungo direzioni tangenti al minimo geometrico. Ciò rende raccomandabile un criterio di arresto anticipato basato sulla geometria.

Poiché la chiusura è consistente per costruzione, l'errore residuo misurato riflette esclusivamente il contenuto informativo delle osservazioni e la geometria di partenza, e non una discrepanza di modello: questo studio costituisce quindi la base pulita contro cui leggere le penalità di inconsistenza di chiusura documentate nello studio parallelo referenziato su dati LES.

9.3 Considerazioni metodologiche trasversali

I due casi di studio, pur nella loro diversità di regime e di geometria, convergono su alcune lezioni comuni. La mal-posizione del problema inverso si manifesta in modo ricorrente come disaccoppiamento tra obiettivo e geometria e come molteplicità di soluzioni quasi equivalenti nelle regioni in cui la firma di scia è debole. Il fattore limitante della ricostruzione non è la *quantità* di dati, ma il loro *contenuto informativo*: aggiungere sensori giova solo dove il segnale è ben condizionato e ricco di struttura, mentre è inefficace dove il campo è instabile o dove la firma è già dispersa. Infine, in entrambi i casi si è confermato che il valore della funzione obiettivo non è un surrogato affidabile della qualità geometrica, e che una metrica di forma indipendente è indispensabile sia per la valutazione sia, idealmente, per il criterio di arresto.

9.4 Limitazioni

I risultati vanno letti alla luce di alcune limitazioni intrinseche all'impostazione adottata. Le osservazioni sono sintetiche e prive di rumore: la natura di *inverse crime*, indispensabile per isolare le variabili di interesse, rende possibili ricostruzioni da firme di scia estremamente tenui (come i probes a oltre venti raggi) che in un contesto sperimentale, dominato dal rumore di misura, non sarebbero recuperabili. Ogni caso è stato inoltre studiato a un singolo numero di Reynolds e a una singola condizione operativa; il profilo alare è stato analizzato solo a incidenza nulla, cioè nella condizione di segnale di scia più debole e simmetrico. La parametrizzazione geometrica è stata mantenuta fissa (FFD a poche stazioni), il che introduce un limite strutturale di accuratezza non riducibile agendo sulla sola misura. Infine, il flusso è stato assunto stazionario, escludendo i regimi instazionari e transizionali.

9.5 Sviluppi futuri

Le limitazioni appena elencate indicano naturalmente le direzioni di sviluppo più promettenti.

La più immediata è l'introduzione di *rumore di misura* nelle osservazioni e la valutazione della robustezza del metodo al degradarsi del rapporto segnale-rumore, passo indispensabile per avvicinare lo studio a un impiego sperimentale con dati PIV o LDV. In quel contesto, le finestre di scia prossima e intermedia (dove il contrasto geometrico è più intenso) sono quelle che manterrebbero significato, il che ne rafforza l'interesse pratico.

Una seconda direzione riguarda l'*arricchimento del segnale di osservazione*. Per il profilo alare, l'estensione a uno o più angoli di attacco non nulli aumenterebbe l'asimmetria di quantità di moto trasversale nella scia e, con essa, il *condizionamento* del recupero del camber, che a incidenza nulla è osservabile ma governato da un gradiente debole in prossimità dell'ottimo, potenzialmente estendendo anche la finestra utile a valle. Più in generale, l'impiego combinato di finestre multiple suggerito dalla complementarità osservata tra scia intermedia e lontana nel caso tozzo, un arricchimento della parametrizzazione FFD, o l'aggiunta di termini obiettivo sensibili alla portanza, potrebbero migliorare il condizionamento del problema ed evitare gli arresti in minimi locali.

Sul piano algoritmico, la questione del criterio di arresto merita un'analisi dedicata, perché intreccia due obiettivi distinti. Il primo è di pura *efficienza*: poiché il grosso della ricostruzione avviene nelle prime dieci-quindici iterazioni e la lunga coda asintotica affina il

punto stazionario KKT a costo CFD elevato ma con guadagno geometrico trascurabile, un arresto sul *plateau numerico dell'obiettivo* (interruzione quando la variazione relativa di f resta sotto una soglia, ad esempio l'1%, per un certo numero di iterazioni consecutive) taglierebbe la coda senza penalizzare la qualità, ed è sicuro nei casi ben posti, in cui la convergenza è monotona. Il secondo obiettivo, più delicato, è *evitare il degrado* osservato nel caso NACA0015/W1, dove la forma migliore viene attraversata a metà processo mentre f sta ancora scendendo: qui un criterio basato sul solo f è per costruzione inefficace, perché arresterebbe l'ottimizzazione in prossimità del minimo dell'obiettivo, cioè proprio nel punto geometricamente peggiore. Un arresto basato sull'errore geometrico coglierebbe la forma corretta, ma non è disponibile in applicazione, dove la geometria vera è ignota. Restano percorribili due surrogati che non la richiedono: un *early stopping su un sottoinsieme di validazione* dei probes, che interrompe l'ottimizzazione quando l'obiettivo valutato sui sensori esclusi dal ciclo cessa di migliorare (segnale che l'ottimizzatore sta adattando la scia alla parametrizzazione anziché recuperare la forma); e il monitoraggio della *sensibilità del passo* $\|\Delta \mathbf{x}\|/\Delta f$, il cui aumento repentino segnala l'ingresso nella valle piatta e mal condizionata lungo la quale la geometria può derivare. L'adozione combinata di un criterio di plateau, per l'efficienza, e di uno di questi surrogati, per la sicurezza geometrica, appare la via più promettente per risolvere in pratica la disconnessione KKT/geometria osservata. Parallelamente, l'adozione di parametrizzazioni geometriche più ricche (FFD con più stazioni, o schemi adattivi) sposterebbe il limite strutturale di accuratezza, al prezzo di uno spazio di progetto più ampio da bilanciare con la stabilità dell'ottimizzazione.

Infine, l'estensione a *regimi più complessi*, come numeri di Reynolds più elevati, flussi instazionari con distacco di vortici, o modelli di transizione, e a target geometrici tridimensionali costituisce il naturale banco di prova per la generalità del metodo, riavvicinandolo alle configurazioni di reale interesse ingegneristico. In tale prospettiva, la questione della consistenza della chiusura turbolenta, isolata in questa tesi mediante l'*inverse crime*, tornerebbe centrale e dovrebbe essere affrontata congiuntamente al problema di osservazione qui caratterizzato.

Appendice A

Script di automazione dell'ottimizzazione: runScript.py

Lo script Python seguente orchestra l'intero ciclo di ottimizzazione tramite OpenMDAO (si veda la Sezione 5.1): definisce il modello multidisciplinare, le variabili di progetto (le funzioni di forma FFD simmetriche e la velocità di ingresso `patchV`), la funzione obiettivo `VarianceScaled` e i parametri dell'ottimizzatore IPOPT, e lancia l'ottimizzazione. La velocità effettiva di flusso libero e l'angolo di attacco del caso ($U_\infty = 1$, $\alpha = 0$) sono imposti proprio attraverso il design variable `patchV`.

```

1  #!/usr/bin/env python
2  """
3  DAfoam run script for the cylinder at low-speed
4  """
5
6  # =====
7  # Imports
8  # =====
9  import os
10 import argparse
11 import numpy as np
12 import json
13 from mpi4py import MPI
14 import openmdao.api as om
15 from mphys.multipoint import Multipoint
16 from dafoam.mphys import DAfoamBuilder, OptFuncs
17 from mphys.scenario_aerodynamic import ScenarioAerodynamic
18 from pygeo.mphys import OM_DVGEOCOMP
19
20
21 parser = argparse.ArgumentParser()
22 # which optimizer to use. Options are: IPOPT (default), SLSQP, and SNOPT
23 parser.add_argument("-optimizer", help="optimizer to use", type=str, default="
    IPOPT")
24 # which task to run. Options are: run_driver (default), run_model, compute_totals
    , check_totals
25 parser.add_argument("-task", help="type of run to do", type=str, default="
    run_driver")
26 args = parser.parse_args()
27
28 # =====
29 # Input Parameters
30 # =====
31 U0 = 1.0
32 p0 = 0.0
33 nuTilda0 = 0.0 #1e-12
34 nuphysical = 0.05
35 CL_target = 0 #0.5
36 aoa0 = 0.0
37 A0 = 0.1
38 # rho is used for normalizing CD and CL
39 rho0 = 1.0

```

```

40
41 with open("./probePointCoords.json") as f: #probePointCoords
42     probePointCoords_circ
43     probePointCoords = json.load(f)
44
45 # Input parameters for DAfoam
46 daOptions = {
47     "designSurfaces": ["cylinder"],
48     "solverName": "DASimpleFoam",
49     "useAD": {"mode": "reverse"},
50     "primalMinResTol": 1.2e-13,
51     "primalBC": {
52         "U0": {"variable": "U", "patches": ["farField"], "value": [U0, 0.0,
53             0.0]},
54         "p0": {"variable": "p", "patches": ["farField"], "value": [p0]},
55         "nuTilda0": {"variable": "nuTilda", "patches": ["farField"], "value": [
56             nuTilda0]},
57         "useWallFunction": False,
58     },
59     "function": {
60         "Var": {
61             "type": "variance",
62             "source": "boxToCell",
63             "min": [-20.0, -20.0, -10.0],
64             "max": [20.0, 20.0, 10.0],
65             "indices": [0, 1],
66             "scale": 10.0,
67             "mode": "probePoint",
68             "probePointCoords": probePointCoords["probePointCoords"],
69             "varName": "U",
70             "data": "UData",
71             "varType": "vector",
72             "components": [0],
73             "timeDependentRefData": False,
74             "addToAdjoint": True,
75         },
76     },
77     "adjEqnOption": {"gmresRelTol": 1.0e-13, "pcFillLevel": 1, "jacMatReOrdering"
78         : "rcm", "gmresMaxIters": 5000,
79         "gmresRestart": 150,},
80     "normalizeStates": {
81         "U": U0,
82         "p": U0 * U0 / 2.0,
83         #"nuTilda": nuTilda0 * 10.0,
84         "phi": 1.0,
85     },
86
87     "inputInfo": {
88         "aero_vol_coords": {"type": "volCoord", "components": ["solver", "
89             function"]},
90         "patchV": {
91             "type": "patchVelocity",
92             "patches": ["farField"],
93             "flowAxis": "x",
94             "normalAxis": "y",
95             "components": ["solver", "function"],

```

```

91         },
92     },
93 }
94
95 # Mesh deformation setup
96 meshOptions = {
97     "useRotations": True,
98     "evalMode": "exact",
99     "gridFile": os.getcwd(),
100     "fileType": "OpenFOAM",
101     # point and normal for the symmetry plane
102     "symmetryPlanes": [[[0.0, 0.0, 0.0], [0.0, 0.0, 1.0]],
103                        [[0.0, 0.0, 1.0], [0.0, 0.0, 1.0]]],
104 }
105
106 # Top class to setup the optimization problem
107 class Top(Multipoint):
108     def setup(self):
109
110         # create the builder to initialize the DASolvers
111         dafoam_builder = DAFoamBuilder(daOptions, meshOptions, scenario="
aerodynamic")
112         dafoam_builder.initialize(self.comm)
113
114         # add the design variable component to keep the top level design
variables
115         self.add_subsystem("dvs", om.IndepVarComp(), promotes=["*"])
116
117         # add the mesh component
118         self.add_subsystem("mesh", dafoam_builder.get_mesh_coordinate_subsystem()
)
119
120         # add the geometry component (FFD)
121         self.add_subsystem("geometry", OM_DVGEOCOMP(file="FFD/cylinderFFD.xyz",
type="ffd"))
122
123         # add a scenario (flow condition) for optimization, we pass the builder
124         # to the scenario to actually run the flow and adjoint
125         self.mphys_add_scenario("scenario1", ScenarioAerodynamic(aero_builder=
dafoam_builder))
126
127         # need to manually connect the x_aero0 between the mesh and geometry
components
128         # here x_aero0 means the surface coordinates of structurally undeformed
mesh
129         self.connect("mesh.x_aero0", "geometry.x_aero_in")
130         # need to manually connect the x_aero0 between the geometry component and
the scenario1
131         # scenario group
132         self.connect("geometry.x_aero0", "scenario1.x_aero")
133
134     def configure(self):
135         super().configure()
136
137         # 1. Coordinate
138         points = self.mesh.mphys_get_surface_mesh()

```

```

139     self.geometry.nom_add_discipline_coords("aero", points)
140
141     # 2. Indici FFD
142     pts = self.geometry.DVGeo.getLocalIndex(0)
143
144     dir_y = np.array([0.0, 1.0, 0.0]) # Direzione Y
145     shapes_symm = [] # Unica lista per deformazione simmetrica
146
147     n_i = pts.shape[0] # 5 (punti in X)
148
149     for i in range(n_i):
150         # NESSUN BLOCCO: Lasciamo muovere anche Leading Edge (0) e Trailing
151         Edge (4)
152         # per permettere di arrotondare la punta come un cilindro
153
154         # Punti BOTTOM (j=0) si muovono verso il basso (-dir_y)
155         # Punti TOP (j=1) si muovono verso l'alto (+dir_y)
156         # Così un valore positivo della DV "gonfia" l'ellisse simmetricamente
157
158         shapes_symm.append({
159             pts[i, 0, 0]: -dir_y, pts[i, 0, 1]: -dir_y, # Fila inferiore
160             pts[i, 1, 0]: dir_y, pts[i, 1, 1]: dir_y # Fila superiore
161         })
162
163         # Registra UN SOLO gruppo di variabili
164         self.geometry.nom_addShapeFunctionDV(dvName="shape_symm", shapes=
165         shapes_symm)
166
167         # Inizializza i valori
168         n_symm = len(shapes_symm)
169         self.dvs.add_output("shape_symm", val=np.zeros(n_symm))
170         self.dvs.add_output("patchV", val=np.array([U0, aoa0]))
171
172         # Connessioni
173         self.connect("shape_symm", "geometry.shape_symm")
174         self.connect("patchV", "scenario1.patchV")
175
176         # 6. Definizione Design Variables
177         self.add_design_var("shape_symm", lower=-1, upper=1, scaler=1.0)
178
179         # 7. Obiettivi
180         self.add_objective("scenario1.aero_post.Var", scaler=10.0)
181
182     # OpenMDAO setup
183     prob = om.Problem()
184     prob.model = Top()
185     prob.setup(mode="rev")
186     om.n2(prob, show_browser=False, outfile="mphys.html")
187
188     # initialize the optimization function
189     optFuncs = OptFuncs(daOptions, prob)
190
191     # use pyoptsparse to setup optimization
192     prob.driver = om.pyOptSparseDriver()
193     prob.driver.options["optimizer"] = args.optimizer
194     # options for optimizers

```

```

192 if args.optimizer == "SNOPT":
193     prob.driver.opt_settings = {
194         "Major feasibility tolerance": 1.0e-7,
195         "Major optimality tolerance": 1.0e-7,
196         "Minor feasibility tolerance": 1.0e-7,
197         "Verify level": -1,
198         "Function precision": 1.0e-5,
199         "Major iterations limit": 100,
200         "Nonderivative linesearch": None,
201         "Print file": "opt_SNOPT_print.txt",
202         "Summary file": "opt_SNOPT_summary.txt",
203     }
204 elif args.optimizer == "IPOPT":
205     prob.driver.opt_settings = {
206         "tol": 1.0e-7,
207         "constr_viol_tol": 1.0e-7,
208         "max_iter": 200,
209         "print_level": 5,
210         "output_file": "opt_IPOPT.txt",
211         "mu_strategy": "adaptive",
212         "limited_memory_max_history": 30,
213         "nlp_scaling_method": "gradient-based", # "none"
214         "alpha_for_y": "full",
215         "recalc_y": "yes",
216         "alpha_red_factor": 0.5,
217         "line_search_method": "filter",
218         "accept_every_trial_step": "no", # Accettare step troppo conservativi
219         "watchdog_shortened_iter_trigger": 5, # Aumenta esplorazione
220         "max_soc": 2, # Second-order correction steps
221         "nu_init": 1e-1, # Barrier parameter iniziale più permissivo
222         # "nu_inc": 10.0, # Aumenta barriera più aggressivamente
223         "adaptive_mu_globalization": "kkt-error",
224     }
225 elif args.optimizer == "SLSQP":
226     prob.driver.opt_settings = {
227         "ACC": 1.0e-7,
228         "MAXIT": 150,
229         "IFILE": "opt_SLSQP.txt",
230     }
231 else:
232     print("optimizer arg not valid!")
233     exit(1)
234
235 prob.driver.options["debug_print"] = ["nl_cons", "objs", "desvars"]
236 prob.driver.options["print_opt_prob"] = True
237 prob.driver.hist_file = "OptView.hst"
238
239 if args.task == "run_driver":
240     prob.run_driver()
241     final_shape_symm = prob.get_val("dvs.shape_symm")
242     np.savetxt("shape_state.txt", final_shape_symm)
243
244 elif args.task == "run_model":
245     # just run the primal once
246     prob.run_model()
247     print("patchV: ", prob.get_val("scenario1.patchV"))

```

```

248 elif args.task == "compute_totals":
249     # just run the primal and adjoint once
250     prob.run_model()
251     totals = prob.compute_totals()
252     if MPI.COMM_WORLD.rank == 0:
253         print(totals)
254 elif args.task == "check_totals":
255     # verify the total derivatives against the finite-difference
256     prob.run_model()
257     prob.check_totals(compact_print=False, step=1e-3, form="central", step_calc="
abs")
258 else:
259     print("task arg not found!")
260     exit(1)
261 # =====
262 # PLOT DELLA FUNZIONE OBIETTIVO IPOPT
263
264 import matplotlib.pyplot as plt
265
266 def parse_ipopt_log(filename):
267     iters = []
268     objs = []
269
270     if not os.path.exists(filename):
271         print(f"[WARNING] File {filename} non trovato.")
272         return iters, objs
273
274     with open(filename, "r") as f:
275         for line in f:
276             # Le righe valide iniziano tipicamente con spazi + numero iterazione
277             parts = line.strip().split()
278             if len(parts) < 2:
279                 continue
280
281             # Tenta di interpretare la prima colonna come intero (iterazione)
282             try:
283                 iter_idx = int(parts[0])
284             except ValueError:
285                 continue # Non è una riga di iterazione IPOPT
286
287             # La seconda colonna è l'obiettivo
288             try:
289                 obj_val = float(parts[1])
290             except ValueError:
291                 continue
292
293             iters.append(iter_idx)
294             objs.append(obj_val)
295
296     return iters, objs
297
298 logfile = "opt_IPOPT.txt"
299 iters, objs = parse_ipopt_log(logfile)
300
301 if len(objs) > 0:
302     # Salva CSV

```

```

303 np.savetxt(
304     "objective_history.csv",
305     np.column_stack([iters, objs]),
306     delimiter=",",
307     header="iter,objective",
308     comments=""
309 )
310
311 # Plot
312 plt.figure(figsize=(7, 5))
313 plt.plot(iters, objs, marker="o", linewidth=2)
314 plt.xlabel("Iteration")
315 plt.ylabel("Objective Function")
316 plt.title("IPOPT Objective Progression")
317 plt.yscale('log')
318 plt.grid(True)
319 plt.tight_layout()
320 plt.savefig("objective_plot.png", dpi=200)
321 plt.close()
322
323 print("\n[INFO] Plot obiettivo salvato in: objective_plot.png")
324 print("[INFO] CSV salvato in: objective_history.csv\n")
325
326 else:
327     print("\n[WARNING] Nessun dato di obiettivo trovato nel log IPOPT.\n")
328     # =====
329     # PLOT AGGIUNTIVO: ||d|| (Norma del passo) da opt_IPOPT.txt
330     # =====
331     import matplotlib.pyplot as plt
332
333     def plot_inf_du():
334         iterations = []
335         inf_du = []
336
337         try:
338             with open("opt_IPOPT.txt", "r") as f:
339                 for line in f:
340                     parts = line.split()
341                     # Verifica che la riga inizi con un numero
342                     if len(parts) >= 6 and parts[0].isdigit():
343                         try:
344                             it = int(parts[0])
345                             # Prendo infdu
346                             val = float(parts[3])
347
348                             iterations.append(it)
349                             inf_du.append(val)
350                         except ValueError:
351                             continue
352
353                 if not iterations:
354                     print("[WARN] Non sono stati trovati dati in opt_IPOPT.txt")
355                     return
356
357                 # Creazione del grafico
358                 plt.figure(figsize=(10, 6))

```

```

359     plt.plot(iterations, inf_du, '-o', color='purple', linewidth=1.5,
360             markersize=4)
361
362     plt.xlabel('Iterazioni', fontsize=12)
363     plt.ylabel('inf_du', fontsize=12)
364     plt.title('Vicinanza a punto di ottimo matematico', fontsize=14)
365
366     # SCALA LOGARITMICA
367     plt.yscale('log')
368
369     plt.grid(True, which="both", linestyle='--', alpha=0.6)
370     plt.tight_layout()
371
372     # Salva il file
373     output_name = "convergence_inf_du.png"
374     plt.savefig(output_name, dpi=150)
375     print(f"[INFO] Grafico salvato correttamente: {output_name}")
376
377     except FileNotFoundError:
378         print("[ERROR] Il file opt_IPOPT.txt non è stato trovato nella cartella
379         corrente.")
380
381 # Esegui la funzione
382 plot_inf_du()

```

Appendice B

Calcolo dell'errore geometrico: `geometric_error.py`

Lo script seguente implementa la metrica di valutazione geometrica descritta nella Sezione 6.2: estrae il profilo del corpo dalla mesh deformata dell'ultima iterazione (leggendo direttamente i file `polyMesh/faces` e `polyMesh/points` di OpenFOAM), lo esprime in coordinate polari e calcola l'RMSE, il MAE e l'errore massimo radiali rispetto al cilindro target, producendo il grafico a tre pannelli usato in tutto il Capitolo 6.

```
1  #!/usr/bin/env python3
2  """
3  geometric_error.py
4  -----
5  Produces three plots shown in the "Threshold Development" slide:
6      1. Profiles Comparison (Cartesian x-y)
7      2. Radius as a function of theta (r vs theta, shaded error)
8      3. Squared Error by Angle
9
10
11
12  Run this script from inside the cylinder_adj/ folder:
13      python3 geometric_error.py
14
15
16
17  It reads:
18      - 0.0NNN/polyMesh/points.gz    deformed (optimized) surface mesh
19      - constant/polyMesh/faces.gz    face connectivity (to find cylinder patch)
20
21
22
23  Output: geometric_error.png
24  """
25
26
27
28  import os
29  import gzip
30  import numpy as np
31  import matplotlib
32  matplotlib.use("Agg")
33  import matplotlib.pyplot as plt
34  import matplotlib.gridspec as gridspec
35
36
37
38  #
39  # 1. Load cylinder-patch face point indices
40  #
41  LAST_ITER_FOLDER = None
42  for d in sorted(os.listdir(".")):
43      try:
44          val = float(d)
45          if val > 0 and os.path.exists(os.path.join(d, "polyMesh", "points.gz")):
```

```

46         LAST_ITER_FOLDER = d
47     except ValueError:
48         pass
49
50
51
52 if LAST_ITER_FOLDER is None:
53     raise FileNotFoundError(
54         "No iteration folder with polyMesh/points.gz found. "
55         "Make sure you run this script inside cylinder_adj/."
56     )
57 print(f"[INFO] Using optimised mesh from: {LAST_ITER_FOLDER}/")
58
59
60
61 # Boundary: cylinder patch  nFaces=400, startFace=79600
62 CYL_START_FACE = 79600
63 CYL_N_FACES    = 400
64 TOTAL_FACES    = 160400
65 TOTAL_POINTS   = 80800
66
67
68
69
70
71 def _read_openfoam_faces(path):
72     with gzip.open(path, "rt") as f:
73         content = f.read()
74         lines = content.split("\n")
75         start = None
76         for i, line in enumerate(lines):
77             stripped = line.strip()
78             if stripped.isdigit() and int(stripped) > 10000:
79                 start = i + 2
80                 break
81         faces = []
82         for line in lines[start:]:
83             stripped = line.strip()
84             if stripped.startswith("4("):
85                 faces.append([int(v) for v in stripped[2:-1].split()])
86             if len(faces) == TOTAL_FACES:
87                 break
88         return faces
89
90
91
92
93
94 def _read_openfoam_points(path, n_points):
95     with gzip.open(path, "rt") as f:
96         content = f.read()
97         lines = content.split("\n")
98         start = None
99         for i, line in enumerate(lines):
100             if line.strip() == str(n_points):
101                 start = i + 2

```

```

102         break
103     pts = []
104     for line in lines[start:]:
105         stripped = line.strip()
106         if stripped.startswith("(") and stripped.endswith(")"):
107             pts.append([float(v) for v in stripped[1:-1].split()])
108         if len(pts) == n_points:
109             break
110     return np.array(pts)
111
112
113
114
115
116 print("[INFO] Reading face connectivity ")
117 faces = _read_openfoam_faces(os.path.join("constant", "polyMesh", "faces.gz"))
118 cyl_faces = faces[CYL_START_FACE : CYL_START_FACE + CYL_N_FACES]
119 cyl_pt_indices = sorted({idx for f in cyl_faces for idx in f})
120
121
122
123 print("[INFO] Reading optimised mesh points ")
124 opt_pts_all = _read_openfoam_points(
125     os.path.join(LAST_ITER_FOLDER, "polyMesh", "points.gz"), TOTAL_POINTS
126 )
127
128
129
130 # Keep only z0 plane (one of the two z-layers)
131 opt_cyl_xy = np.array(
132     [opt_pts_all[i, :2] for i in cyl_pt_indices if abs(opt_pts_all[i, 2]) < 0.01]
133 )
134
135
136
137 # Sort by angle
138 theta_opt_rad = np.arctan2(opt_cyl_xy[:, 1], opt_cyl_xy[:, 0])
139 sort_idx = np.argsort(theta_opt_rad)
140 opt_cyl_xy = opt_cyl_xy[sort_idx]
141 theta_deg = theta_opt_rad[sort_idx] * 180.0 / np.pi
142 r_opt = np.sqrt(opt_cyl_xy[:, 0]**2 + opt_cyl_xy[:, 1]**2)
143
144
145
146 # Fix: close the angular loop so the plot spans exactly -180 +180
147 # arctan2 returns (-180, +180], so we may be missing the -180 end.
148 # Duplicate the last point (near +180) as -180 and prepend it,
149 # and duplicate the first point (near -180) as +180 and append it.
150 theta_deg_closed = np.concatenate([[ -180.0], theta_deg, [180.0]])
151 r_opt_closed = np.concatenate([[r_opt[-1]], r_opt, [r_opt[0]]])
152 opt_cyl_xy_closed = np.vstack([opt_cyl_xy[-1:], opt_cyl_xy, opt_cyl_xy[:1]])
153
154
155
156 # Use the closed arrays for plots 2 & 3
157 theta_plot = theta_deg_closed

```

```

158 r_plot      = r_opt_closed
159
160
161
162 #
163 # 2. Geometric error metrics (on original, non-duplicated data)
164 #
165 R_TARGET = 1.0
166 error     = r_opt - R_TARGET
167 sq_error  = error ** 2
168 abs_error = np.abs(error)
169
170
171
172 sq_error_plot = np.concatenate([[sq_error[-1]], sq_error, [sq_error[0]]])
173
174
175
176 MSE = float(np.mean(sq_error))
177 RMSE = float(np.sqrt(MSE))
178 MAE = float(np.mean(abs_error))
179 MAX_e = float(np.max(abs_error))
180 MAX_theta = float(theta_deg[np.argmax(abs_error)])
181
182
183
184 print(f"[RESULT] MSE = {MSE:.4f}")
185 print(f"[RESULT] RMSE = {RMSE:.4f}")
186 print(f"[RESULT] MAE = {MAE:.4f}")
187 print(f"[RESULT] Max = {MAX_e:.4f} @ {MAX_theta:.1f}°")
188
189
190
191 #
192 # 3. Build the figure (3 plots, single row)
193 #
194 fig = plt.figure(figsize=(16, 5))
195 fig.patch.set_facecolor("white")
196
197
198
199 header = (
200     f"GEOMETRIC ERROR | RMSE={RMSE:.4f} MAE={MAE:.4f} "
201     f"Max={MAX_e:.4f} @ {MAX_theta:.1f}°"
202 )
203 fig.suptitle(header, fontsize=20, fontweight="bold", y=1.03)
204
205
206
207 gs = gridspec.GridSpec(
208     1, 3,
209     figure=fig,
210     top=0.90, bottom=0.14,
211     left=0.06, right=0.975,
212     wspace=0.34,
213 )

```

```

214
215
216
217 # Plot 1: Profiles Comparison (Cartesian)
218 ax1 = fig.add_subplot(gs[0, 0])
219
220
221
222 theta_circle = np.linspace(-np.pi, np.pi, 400)
223 ax1.plot(np.cos(theta_circle), np.sin(theta_circle),
224          color="green", linewidth=1.8, label="Cylinder r=1.0")
225 ax1.plot(opt_cyl_xy[:, 0], opt_cyl_xy[:, 1],
226          color="red", linestyle="--", linewidth=1.4, label="Optimized Profile")
227
228
229
230 ax1.set_xlim(-1.2, 1.2)
231 ax1.set_ylim(-1.2, 1.2)
232 ax1.set_aspect("equal")
233 ax1.axhline(0, color="gray", linewidth=0.5)
234 ax1.axvline(0, color="gray", linewidth=0.5)
235 ax1.set_xlabel("x", fontsize=16)
236 ax1.set_ylabel("y", fontsize=16)
237 ax1.set_title("Profiles Comparison", fontsize=17)
238 ax1.tick_params(labelsize=13)
239 ax1.legend(fontsize=12, loc="lower right")
240 ax1.grid(True, linestyle="--", alpha=0.4)
241
242
243
244 # Plot 2: Radius as a function of theta
245 ax2 = fig.add_subplot(gs[0, 1])
246
247
248
249 ax2.plot([-180, 180], [R_TARGET, R_TARGET],
250          color="green", linewidth=1.5, label="r target = 1.0")
251 ax2.plot(theta_plot, r_plot,
252          color="red", linestyle="--", linewidth=1.3, label="r optimized")
253
254
255
256 ax2.fill_between(theta_plot, r_plot, R_TARGET,
257                  where=(r_plot < R_TARGET),
258                  color="salmon", alpha=0.5, label="Radial Error")
259 ax2.fill_between(theta_plot, r_plot, R_TARGET,
260                  where=(r_plot >= R_TARGET),
261                  color="salmon", alpha=0.5)
262
263
264
265 ax2.set_xlabel(" (degrees)", fontsize=16)
266 ax2.set_ylabel("r", fontsize=16)
267 ax2.set_title("Radius as a function of theta", fontsize=17)
268 ax2.set_xlim(-180, 180)
269 ax2.set_xticks(range(-180, 181, 60))

```

```

270 ax2.tick_params(labelsize=13)
271 ax2.legend(fontsize=12)
272 ax2.grid(True, linestyle="--", alpha=0.4)
273
274
275
276 # Plot 3: Squared Error by Angle
277 ax3 = fig.add_subplot(gs[0, 2])
278
279
280
281 ax3.fill_between(theta_plot, sq_error_plot, color="steelblue", alpha=0.75)
282 ax3.plot(theta_plot, sq_error_plot, color="steelblue", linewidth=0.8)
283
284
285
286 ax3.set_xlabel(" (degrees)", fontsize=16)
287 ax3.set_ylabel(r"${r_{opt}} - r_{target}$^2$", fontsize=16)
288 ax3.set_title("Squared Error by Angle", fontsize=17)
289 ax3.set_xlim(-180, 180)
290 ax3.set_xticks(range(-180, 181, 60))
291 ax3.tick_params(labelsize=13)
292 # notazione scientifica: la potenza (es. 1e-4) va in cima all'asse,
293 # così i numeri sull'asse y restano corti e l'etichetta non invade il grafico
    centrale
294 ax3.ticklabel_format(axis="y", style="sci", scilimits=(0, 0))
295 ax3.yaxis.get_offset_text().set_fontsize(13)
296 ax3.grid(True, linestyle="--", alpha=0.4)
297
298
299
300 plt.savefig("geometric_error.png", dpi=180, bbox_inches="tight")
301 print("[INFO] Saved geometric_error.png")
302 plt.close()

```

Appendice C

Post-processing della convergenza: `plottingtot.py`

Lo script seguente genera i grafici della storia della funzione obiettivo e della convergenza dell'infeasibility duale (`inf_du`) a partire dal file di log `opt_IPOPT.txt` prodotto dall'ottimizzatore.

```
1  #!/usr/bin/env python3
2  import os
3  import numpy as np
4
5  # PLOT DELLA FUNZIONE OBIETTIVO IPOPT
6
7  import matplotlib.pyplot as plt
8
9  # Font piu' grandi per leggibilita' quando le figure vengono ridotte nella tesi
10 plt.rcParams.update({
11     "axes.titlesize": 18,
12     "axes.labelsize": 15,
13     "xtick.labelsize": 13,
14     "ytick.labelsize": 13,
15     "legend.fontsize": 13,
16 })
17
18 def parse_ipopt_log(filename):
19     iters = []
20     objs = []
21
22     if not os.path.exists(filename):
23         print(f"[WARNING] File {filename} non trovato.")
24         return iters, objs
25
26     with open(filename, "r") as f:
27         for line in f:
28             # Le righe valide iniziano tipicamente con spazi + numero iterazione
29             parts = line.strip().split()
30             if len(parts) < 2:
31                 continue
32
33             # Tenta di interpretare la prima colonna come intero (iterazione)
34             try:
35                 iter_idx = int(parts[0])
36             except ValueError:
37                 continue # Non è una riga di iterazione IPOPT
38
39             # La seconda colonna è l'obiettivo
40             try:
41                 obj_val = float(parts[1])
42             except ValueError:
43                 continue
44
45             iters.append(iter_idx)
46             objs.append(obj_val)
47
```

```

48     return iters, objs
49
50 logfile = "opt_IPOPT.txt"
51 iters, objs = parse_ipopt_log(logfile)
52
53 if len(objs) > 0:
54     # Salva CSV
55     np.savetxt(
56         "objective_history.csv",
57         np.column_stack([iters, objs]),
58         delimiter=",",
59         header="iter,objective",
60         comments=""
61     )
62
63     # Plot
64     plt.figure(figsize=(7.5, 5.5))
65     plt.plot(iters, objs, marker="o", linewidth=2)
66     plt.xlabel("Iteration")
67     plt.ylabel("Objective Function")
68     plt.title("IPOPT Objective Progression")
69     plt.yscale('log')
70     plt.grid(True, which="both", linestyle='--', alpha=0.6)
71     plt.grid(True)
72     plt.tight_layout()
73     plt.savefig("objective_plot.png", dpi=200)
74     plt.close()
75
76     print("\n[INFO] Plot obiettivo salvato in: objective_plot.png")
77     print("[INFO] CSV salvato in: objective_history.csv\n")
78
79 else:
80     print("\n[WARNING] Nessun dato di obiettivo trovato nel log IPOPT.\n")
81     # =====
82     # PLOT AGGIUNTIVO: ||d|| (Norma del passo) da opt_IPOPT.txt
83     # =====
84     import matplotlib.pyplot as plt
85
86     def plot_inf_du():
87         iterations = []
88         inf_du = []
89
90         try:
91             with open("opt_IPOPT.txt", "r") as f:
92                 for line in f:
93                     parts = line.split()
94                     # Verifica che la riga inizi con un numero
95                     if len(parts) >= 6 and parts[0].isdigit():
96                         try:
97                             it = int(parts[0])
98                             # Prendo infdu
99                             val = float(parts[3])
100
101                             iterations.append(it)
102                             inf_du.append(val)
103                         except ValueError:

```

```

104         continue
105
106     if not iterations:
107         print("[WARN] Non sono stati trovati dati in opt_IPOPT.txt")
108         return
109
110     # Creazione del grafico
111     plt.figure(figsize=(7.5, 5.5))
112     plt.plot(iterations, inf_du, '-o', color='purple', linewidth=1.5,
113             markersize=4)
114
115     plt.xlabel('Iteration')
116     plt.ylabel('inf_du')
117     plt.title('IPOPT Dual Infeasibility (inf_du)')
118
119     # SCALA LOGARITMICA
120     plt.yscale('log')
121
122     plt.grid(True, which="both", linestyle='--', alpha=0.6)
123     plt.tight_layout()
124
125     # Salva il file
126     output_name = "convergence_inf_du.png"
127     plt.savefig(output_name, dpi=150)
128     print(f"[INFO] Grafico salvato correttamente: {output_name}")
129
130 except FileNotFoundError:
131     print("[ERROR] Il file opt_IPOPT.txt non è stato trovato nella cartella
    corrente.")
    plot_inf_du()

```

Appendice D

Campo di velocità e condizioni al contorno: file 0/U

Il file OpenFOAM seguente definisce il campo di velocità iniziale e le condizioni al contorno del caso di studio (Sezione 5.2): flusso libero $U_\infty = 1$ in direzione x ad angolo di attacco nullo, condizione di *no-slip* sulla superficie del corpo (*cylinder*), condizione *inletOutlet* sul contorno esterno (*farField*) e simmetria sulle facce che riducono il dominio a una slice pseudo-2D (*backAndFront*).

```

1  /*-----*- C++ -*-----*\
2  | ===== |
3  |  \ \  /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O peration   | Version:  v1812                      |
5  |  \ \  /  A nd         | Web:      www.OpenFOAM.com           |
6  |  \ \  /  M anipulation |                                     |
7  |-----*\
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object        U;
15 }
16 // * * * * *
17
18 dimensions      [0 1 -1 0 0 0];
19
20
21 internalField    uniform (1 0 0);
22
23 boundaryField
24 {
25     cylinder
26     {
27         type      fixedValue;
28         value      uniform (0 0 0);
29     }
30     backAndFront
31     {
32         type      symmetry;
33     }
34     farField
35     {
36         type      inletOutlet;
37         inletValue uniform (1 0 0);
38         value      uniform (1 0 0);
39     }
40 }
41
42
43 // *****

```

Bibliografia

- [1] Z. Zhang and G. E. Karniadakis. Effect of turbulence-closure consistency on airfoil identification. *Computers & Fluids*, page 107155, 2026. arXiv:2511.08341.
- [2] M. Drela. XFOIL: An analysis and design system for low reynolds number airfoils. *Lecture Notes in Engineering*, 54:1–12, 1989.
- [3] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.
- [4] O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation – part I: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113(478):1311–1328, 1987.
- [5] G. Artana, A. Cammilleri, J. Carlier, and E. Mémin. Reconstruction of unsteady viscous flows using data assimilation schemes. *Journal of Computational Physics*, 316:255–280, 2016.
- [6] D. P. G. Foures, C. P. Caulfield, and P. J. Schmid. Variational framework for flow optimization using semi-norm constraints. *Physical Review E*, 86(2):026306, 2014.
- [7] K. Sasaki, R. Vinuesa, A. V. G. Cavalieri, P. Schlatter, and P. Jordan. Transfer functions for flow predictions in wall-bounded turbulence. *Journal of Fluid Mechanics*, 2024.
- [8] M. Iglesias, Y. Lu, and A. Stuart. A bayesian level set method for geometric inverse problems. *Interfaces and Free Boundaries*, 18(2):181–217, 2016.
- [9] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64:525–545, 2019.
- [10] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [11] X. Jin, S. Cai, H. Li, and G. E. Karniadakis. NSFnets (Navier–Stokes flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 426:109951, 2021.
- [12] L. Sun, H. Gao, S. Pan, and J.-X. Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361:112732, 2020.
- [13] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, Berlin, 1971.
- [14] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

- [15] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.
- [16] A. Jameson, N. A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier–Stokes equations. In *35th AIAA Aerospace Sciences Meeting and Exhibit*, 1998.
- [17] E. J. Nielsen and W. K. Anderson. Aerodynamic design optimization on unstructured meshes using the Navier–Stokes equations. In *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1999.
- [18] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *ACM SIGGRAPH Computer Graphics*, volume 20, pages 151–160, 1986.
- [19] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631, 1998.
- [20] P. He, C. A. Mader, J. R. R. A. Martins, and K. J. Maki. DAfoam: An open-source adjoint framework for multidisciplinary design optimization with OpenFOAM. *AIAA Journal*, 58(3):1304–1319, 2020.
- [21] T. D. Economou, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016.
- [22] G. K. W. Kenway, C. A. Mader, P. He, and J. R. R. A. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, 2019.
- [23] J. Anibal, P. He, and J. R. R. A. Martins. Rapid jacobian assembly for the adjoint method in OpenFOAM using algorithmic differentiation. *Computers & Fluids*, 280:106356, 2024.
- [24] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, 2003.
- [25] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- [26] D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.
- [27] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. In *Structural and Multidisciplinary Optimization*, volume 59, pages 1075–1104, 2019.

- [28] D. Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt, Köln, Germany, 1988.
- [29] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.
- [30] Flavio Giannetti and Paolo Luchini. Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics*, 581:167–197, 2007.
- [31] Olivier Marquet, Denis Sipp, and Laurent Jacquin. Sensitivity analysis and passive control of cylinder flow. *Journal of Fluid Mechanics*, 615:221–252, 2008.
- [32] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. Technical Report AIAA Paper 1992-0439, AIAA, 1992.
- [33] P. He, C. A. Mader, J. R. R. A. Martins, and K. J. Maki. An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM. *Computers & Fluids*, 168:285–303, 2018.