

## **UNIVERSITÀ DEGLI STUDI DI PADOVA**

Tesi magistrale facoltà di ingegneria meccanica

# **Analisi fluidodinamica del profilo alare Clark-Y ed ottimizzazione multi-obiettivo tramite algoritmo genetico**

Relatore: Ch. Prof. Ernesto Benini

Correlatore: Ing. Antonello Cherubini

Laureando: Andrea Venturato

Marticola: 607493

**Anno Accademico 2012 - 2013**



*Ai miei Genitori ,parenti ed amici.*



# Indice

ABSTRACT.....	9
1. KiteGen: Eolico Troposferico.....	10
1.1 Kite steering unit.....	14
2. CFD: Computational Fluid Dynamics.....	15
2.1 RANS: Raynolds Averaged Navier-Stokes Equation.....	16
2.2 Equazioni di Navier-Stokes.....	19
2.3 Turbolenza.....	20
2.4 Modelli di turbolenza.....	21
2.4.1 Spalart Allamaras.....	21
2.4.2 Modello k- $\epsilon$ .....	22
2.4.3 Modello k- $\omega$ .....	22
3. Algoritmi Genetici.....	24
3.1 Il modello di Holland.....	24
3.1.1 Selezione.....	25
3.1.2 Crossover.....	26
3.1.3 Mutazione.....	27
3.1.4 Inversione.....	28
3.2 Algoritmo genetico multi obbiettivo.....	29
3.2.1 Ordinamento di Pareto: Goldberg.....	30

3.3 Metodi di preservazione della diversità genetica.....	31
3.3.1 Fitness sharing .....	32
3.3.2 Crowding.....	33
3.3.3 Accoppiamento ristretto .....	34
4. Parametrizzazione del profilo Clark-y.....	35
4.1 Creazione del dominio di calcolo .....	37
4.2 Importazione dei punti del profilo.....	40
4.3 Mesh 2D .....	43
5. Validazione del modello fisico .....	47
5.3 Modello numerico.....	48
6. Ottimizzazione .....	53
6.1 La funzione KITGEN.m.....	54
6.2 Impostazioni dell' algoritmo Genetico.....	57
6.3 Fronte di pareto .....	59
6.4 Soluzione ottimale .....	60
6.5 Confronto .....	62
7. Conclusioni e sviluppi .....	69
Appendice A: PARAM.m .....	71
Appendice B: KITGEN.m .....	73

Appendice C: Routine accessorie.....82

Bibliografia .....83



# ABSTRACT

L'obbiettivo di questa tesi è quello di ottimizzare in termini di efficienza il profilo aerodinamico dell'ala del CarboKite della KiteGen tramite l'algoritmo genetico multi-obbiettivo di MATLAB student.

Per conseguire l'obbiettivo prefissato è stata creata una griglia di calcolo strutturata a celle rettangolari ( C-Mesh ) per la geometria del profilo Clark-y tramite Gambit 2.4.6. Si è proceduto con l'infittimento della mesh di primo tentativo al fine ottenere risultati attendibili per la resistenza aerodinamica (  $0 < y^+ < 1$  ) e per la portanza simulando la griglia di calcolo tramite Student FLUENT dove le equazioni di Navier-Stokes per la corrente fluida sono state risolte nel dominio di calcolo approssimandole con le RANS (Raynolds-Averaged-Navier-Stokes).

Dopo aver raggiunto un'accuratezza sufficiente è stato validato il modello fisico tramite confronto diretto tra le curve polari provenienti dalle simulazioni numeriche ed i dati sperimentali tabulati; inoltre sono stati scritti i file Journal di Gambit e Fluent al fine di automatizzare le procedure di creazione della griglia di calcolo e simulazione fluidodinamica.

La parametrizzazione tramite curve di Bezier della geometria del profilo ha permesso di renderlo agevolmente modificabile in maniera automatica ed al contempo di diminuire il numero di variabili necessarie all'ottimizzazione.

La funzione obbiettivo dell' algoritmo genetico ha provveduto a generare le differenti geometrie ed a simularle per un range di angoli di attacco compresi tra  $-5^\circ$  e  $10^\circ$  al fine di aumentare l' efficienza aerodinamica ed avere allo stesso tempo una curva polare dal profilo piatto.

## 1. KiteGen: Eolico Troposferico



Questo capitolo introduce i concetti base, le possibili configurazioni e le fasi del ciclo di generazione del kitegen.

L' idea del kitegen è di utilizzare un aquilone, vincolato a terra da due cavi, per estrarre energia dai venti di quota rispetto a quelli utilizzati dalle tecnologie eoliche attuali.

La traiettoria degli aquiloni è guidata da un unità di controllo automatico capace di controllare il dispositivo tirando indipendentemente le linee. L' energia del vento è raccolta al suolo convertendo la forza di trazione dell' aquilone in energia elettrica tramite opportuni meccanismi rotanti e generatori elettrici.

Gli aquiloni sono in grado di sfruttare venti ad altitudini molto maggiori rispetto alle torri eoliche(fino a 1000 metri con cavi di 1200-1400m) dove venti di intensità e disponibilità maggiore possono essere trovati praticamente ovunque.

L'idea chiave del progetto KiteGen è di raccogliere l'energia dei venti di alta quota in maniera efficiente in termini di spazio occupato e costi della struttura. Nelle attuali torri eoliche l'estremità delle pale contribuisce all'80% della totalità di energia generata. La ragione principale è che la parte periferica della pala ha una velocità tangenziale maggiore(e, conseguentemente, viene investita da una corrente relativa più intensa) e la potenza estraibile dal vento varia col cubo della velocità. Inoltre la torre e la parte più interna delle pale, non contribuiscono direttamente alla produzione di energia determinando la maggior parte dei costi ed imponendo un limite all'altezza di esercizio.

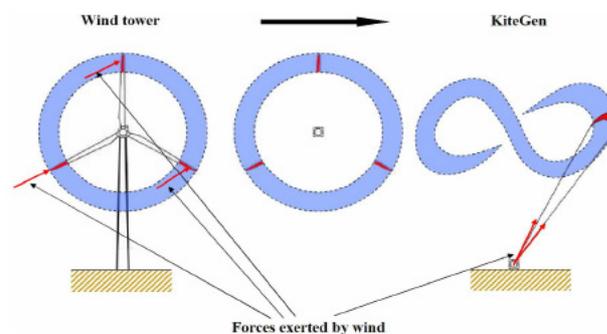


Figura 1 concetto base del ciclo produttivo

Nel KiteGen viene impiegato un aquilone veloce e leggero che fa le veci delle parti terminali delle pale eoliche vincolato a terra da due soli cavi che sostituiscono la più costosa e problematica torre.

Elevata efficienza, manovrabilità, resistenza a trazione e leggerezza sono le caratteristiche principali che un ala deve soddisfare per la produzione di energia sfruttando i venti di alta quota.

L'efficienza aerodinamica è definita come il rapporto tra i coefficienti di lift e drag dell'ala rispettivamente  $CL$  e  $CD$ . Tali coefficienti sono funzione dell'angolo di attacco definito come l'angolo tra l'asse longitudinale del profilo e la direzione della corrente relativa che lo investe.

Nel caso di un'apertura alare infinita  $CL$  e  $CD$  dipendono solo dalla geometria del profilo, nel caso invece che essa sia finita gli effetti della turbolenza sui bordi laterali dell'ala riducono la sua efficienza aerodinamica. Tale perdita di resistenza è tanto maggiore quanto minore è il rapporto  $W/c$  tra l'apertura alare e la corda del profilo. Essendo che, in prima approssimazione, la potenza generata è direttamente proporzionale al quadrato del rapporto  $W/c$  (detto anche aspect ratio) esso dovrebbe essere massimizzato.

La manovrabilità dell'ala, in termini di raggio minimo di traiettoria  $R_f$  dipende anch'esso dal valore dell'apertura.

Essendo la traiettoria ottimale un ciclo a forma di "otto", l'apertura alare dovrebbe essere sufficientemente contenuta per poter affiancare il maggior numero di dispositivi in un'area relativamente piccola. Efficienza e manovrabilità hanno dunque requisiti opposti per quanto riguarda la geometria dell'ala. Per quanto concerne resistenza e leggerezza dipendono più dai materiali utilizzati che da aspetti geometrici. Durante lo sviluppo del KiteGen sono stati utilizzati materiali

flessibili e strutture gonfiabili essendo economiche leggere e sufficientemente rigide.

La forza di portanza dell' aquilone è convertita in potenza meccanica attraverso le forze di trazione che agiscono sulle linee. Dunque, quest' ultime devono essere sufficientemente resistenti per poter sopportare carichi elevati. Allo stesso tempo i cavi devono essere leggeri ed il loro diametro deve essere il minore possibile al fine di limitarne il peso e la resistenza aereaodinamica.

Nel progetto del KiteGen sono state utilizzate linee in materiale composito 8-10 volte più resistenti rispetto ai cavi d' acciaio di pari massa.

Per estrarre energia da venti tra i 200 ed i 1000 m sono necessarie linee tra i 500-1500m ed il prototipo del KiteGen ne dispone di 1000 m.



Figura 2 cavi in materiale composito

## 1.1 Kite steering unit

A livello del terreno , le linee del kite sono avvolte attorno a due tamburi collegati a due motori elettrici i quali sono in grado di funzionare anche da generatori. La traiettoria dell' ala è monitorata utilizzando sia strumentazione wireless direttamente installata a bordo ( GPS, accelerometri) sia con dispositivi a terra, per misurare velocità e posizione del kite, la potenza in uscita, la tensione dei cavi e velocità e direzione del vento. Queste variabili sono utilizzate come feedback da un sistema di controllo elettronico per modificare la traiettoria del kite in tempo reale potendo controllare separatamente la tensione dei cavi.

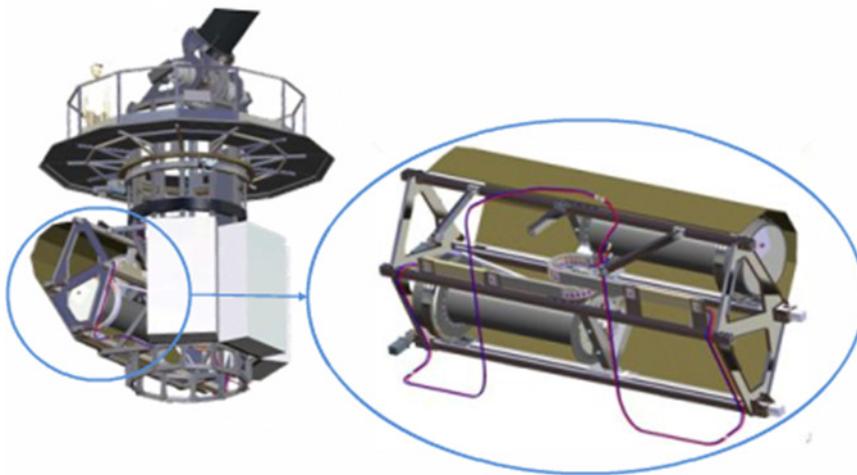


Figura 3 motori/alternatori della kite steering unit

Il sistema, comprensivo di motori/generatori elettrici , tamburi , sensori di bordo e di terra e tutto l' hardware necessario al controllo del kite è definito Kite Steering Unit (KSU), tale sistema nel KiteGen Stem viene utilizzato per manovrare l' ala in un ciclo di produzione detto "KG-yoyo" nel quale la KSU è vincolata a terra e l' energia

è raccolta tramite lo srotolamento progressivo dei cavi che fanno girare l'alternatore calettato assieme ai tamburi.

## **2. CFD: Computational Fluid Dynamics**

La fluidodinamica computazionale (brevemente detta CFD, Computational Fluid Dynamics) è la tecnica che permette lo studio dei problemi di fluidodinamica mediante l'utilizzo del computer. Il principale utilizzo della CFD è quello di risolvere le equazioni di Navier-Stokes e le equazioni ad esse collegate. La risoluzione per via analitica di queste equazioni è fattibile solamente in casi semplici con flussi laminari, e geometrie semplici (sfere, lastre piane), mentre le risoluzioni di casi reali, in cui compaiono di frequente flussi turbolenti, richiedono per forza un approccio numerico. Esistono quindi diversi metodi per risolvere le equazioni di Navier-Stokes, e poiché generalmente sono operazioni dall'elevato costo computazionale, si sono sviluppati approcci via via più raffinati basati su modelli:

Direct Numerical Solution (Soluzione Numerica Diretta: DNS): è l'approccio concettualmente più semplice, si discretizzano lo spazio e il tempo con griglie della dimensione voluta e si eseguono i calcoli su esse. È l'approccio che restituisce i risultati più accurati ma ha un costo computazionale elevatissimo, devono essere impiegati dei super-computer. Per le applicazioni industriali risulta quindi troppo dispendioso.

Reynolds Averaged Navier-Stokes (Equazioni mediate alla Reynolds: RANS): si basano sull'assunzione che si possa vedere il moto turbolento come formato da un

moto medio e da una sua fluttuazione nel tempo. Le grandezze delle equazioni di partenza vengono mediate in un certo intervallo di tempo; così facendo i tempi di calcolo vengono notevolmente ridotti in quanto le scale del moto medio risultano essere notevolmente maggiori di quelle del moto turbolento. Richiedono l'utilizzo di ulteriori equazioni (ad esempio il modello  $k-\epsilon$ ) per la chiusura del problema.

In molti casi devono essere risolte anche altre equazioni contemporaneamente alle equazioni di Navier-Stokes e alle equazioni del modello di turbolenza. Esse possono comprendere quelle relative alla concentrazione di specie diverse (flusso multicomponente), di reazioni chimiche (flussi reagenti), irraggiamento termico, ecc.. Problemi più complessi richiedono la modellazione di ulteriori fenomenologie, come nel caso dei flussi bifase (ad esempio gas / liquido) o nel caso di fluidi non Newtoniani.

## **2.1 RANS: Raynolds Averaged Navier-Stokes Equation**

Nelle applicazioni pratiche, in generale, la sola conoscenza delle grandezze medie può essere sufficiente alla soluzione del problema relativo ad un flusso turbolento; l'idea di base della tecnica RANS è quella di derivare equazioni per le sole grandezze medie (mediate nel tempo) a partire dalle equazioni di Navier-Stokes, al fine di potere ridurre l'enorme carico computazionale richiesto dalla DNS. Il moto medio, inoltre, può essere sia bidimensionale che stazionario, contrariamente alla DNS che richiede sempre equazioni 3D e non stazionarie (questo può essere particolarmente

vantaggioso nella simulazione di determinate condizioni fluidodinamiche). Le equazioni per il moto medio che si ottengono sono esteticamente simili a quelle di Navier-Stokes fatta eccezione per la divergenza del tensore degli sforzi di Reynolds. Gli sforzi turbolenti di Reynolds sono del tutto fittizi rispetto alla realtà fisica e quantificano il trasferimento di energia cinetica dal moto medio alle strutture turbolente. La differenza fra le equazioni di Navier-Stokes e le RANS non è purtroppo solo estetica in quanto mentre le equazioni di Navier-Stokes costituiscono un problema differenziale chiuso, le RANS non lo sono affatto perché il tensore di Reynolds aggiunge altre 6 incognite al problema. Il problema appena citato è noto come problema di chiusura della Turbolenza il quale, in generale, viene risolto introducendo dei modelli per le fluttuazioni turbolente i quali devono riprodurre l'azione dei termini fluttuanti sul moto medio riconducendo i valori degli sforzi turbolenti a sole grandezze relative al moto medio, e, l'idea comunemente usata per raggiungere tale obiettivo è quella di troncare il numero di equazioni ad un certo ordine e modellare le incognite di ordine superiore con delle relazioni approssimate. L'ostacolo principale alla genesi di appropriati Modelli di Turbolenza è quello di coprire l'intervallo più ampio possibile dello spettro turbolento; in quanto le piccole scale sono isotrope ed universali, infatti, presentano le stesse proprietà indipendentemente dalla direzione di osservazione ed in tutte le correnti turbolente senza vincoli in termini di stazionarietà o meno delle condizioni al contorno. Le strutture turbolente di dimensione maggiore, invece, hanno un comportamento dinamico intimamente connesso con il tipo di corrente turbolenta in esame e ciò è alla base della scarsa generalità dei Modelli di Turbolenza che si

prefiggono l'obiettivo di rappresentare l'intero spettro turbolento ed è (il primo macro difetto) uno principali problemi delle simulazioni RANS perché è necessario calibrare il modello (esistono in letteratura dei "magic numbers" che permettono questo).I principali modelli di turbolenza introducono, inoltre, la viscosità cinematica turbolenta, che contrariamente a quella tradizionale, ha la peculiarità di variare sempre da punto a punto del fluido e di essere sempre non stazionaria. L'introduzione della viscosità cinematica turbolenta comporta il notevole vantaggio di spostare l'incognita da un tensore ad uno scalare, però introduce l'allineamento degli autovettori della parte deviatorica del tensore degli sforzi di Reynolds e della parte simmetrica del tensore gradiente della velocità; la proprietà di allineamento degli autovettori dei tensori prima citati non è in alcun modo giustificabile teoricamente e, inoltre, sperimentalmente si è visto che tali autovettori sono per la maggior parte dei casi quasi sempre disallineati. Il disallineamento di cui si è appena accennato produce una differenza fra la soluzione calcolata e misurata e alcune volte è fonte di errori molto grossolani che è il secondo macro difetto delle RANS.I Modelli di Turbolenza basati sulla viscosità cinematica turbolenta sono detti modelli dissipativi e la loro adozione è fisicamente giustificata vista l'intrinseca natura dissipativi della Turbolenza.

In questo capitolo verranno illustrati, sinteticamente, i fondamenti teorici delle RANS (Reynolds Averaged Navier Stokes equations), delle DES (Detached Eddy Simulations) e dei modelli numerici adottati per risolvere il problema di chiusura.

## 2.2 Equazioni di Navier-Stokes

Il moto di un fluido newtoniano non comprimibile è descritto dalle equazioni di Navier Stokes.

$$\frac{\partial u_i}{\partial x_i} = 0$$
$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2}$$

dove  $u_i$  (con  $i = 1, 2, 3$ ) sono le componenti di velocità in un sistema di riferimento cartesiano di coordinate  $x_i$  e  $p$  rappresenta la pressione statica.  $\nu$  e  $\rho$  sono, rispettivamente, la viscosità cinematica e la densità del fluido preso in esame; avendo assunto il fluido come incomprimibile, tali proprietà sono costanti. La risoluzione diretta di queste equazioni è molto costosa computazionalmente parlando; all'aumentare del numero di Reynolds e della complessità del flusso preso in esame, l'approccio diretto (DNS, Direct Numerical Simulation) diventa presto proibitivo e, per questo motivo, è necessario introdurre delle semplificazioni in modo tale da ridurre la complessità delle equazioni e quindi il costo computazionale richiesto per risolverle. A tale scopo si può adottare la scomposizione di Reynolds, grazie alle quale si rappresentano le grandezze che compaiono nelle equazioni di Navier-Stokes come la somma di un valore medio ed una componente fluttuante.

## 2.3 Turbolenza

Lo studio delle correnti viscosi indica che il regime di moto si mantiene laminare soltanto per numeri di Reynolds relativamente piccoli. Al crescere di questo parametro, infatti, il moto assume aspetti profondamente differenti ed i valori delle componenti della velocità e della pressione mostrano rapide variazioni nel tempo, sebbene le condizioni al contorno possano essere (e in gran parte dei casi siano effettivamente) compatibili con un moto stazionario.

Ad esempio, un ipotetico sensore di velocità collocato in un punto fisso di una corrente in media stazionaria (nel senso che risulterà chiaro più avanti) fornisce due andamenti temporali completamente diversi e caratteristici, a seconda che il regime di moto sia laminare oppure turbolento

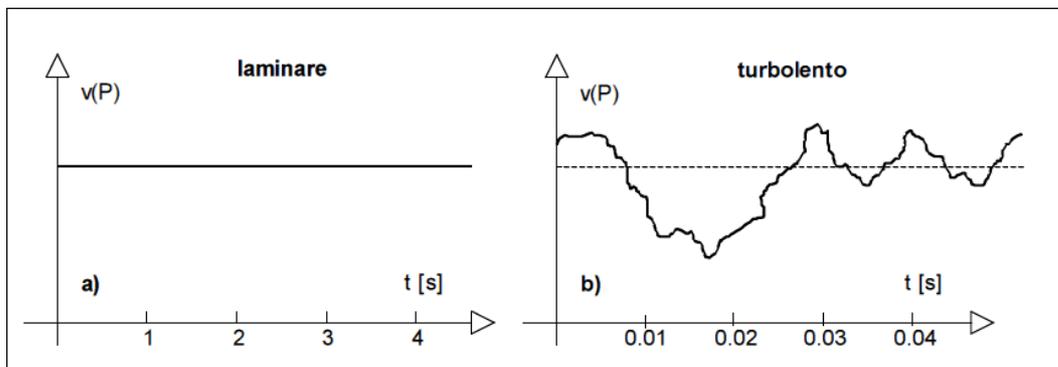


Figura 4 scomposizione in moto medio e fluttuante della velocità

## 2.4 Modelli di turbolenza

Come detto in precedenza i modelli di tipo RANS offrono l'approccio più economico a livello computazionale per il calcolo di flussi industriali turbolenti complessi.

Tipici esempi di questi modelli sono il  $k-\epsilon$  o il  $k-\omega$  nelle loro differenti formulazioni. Questi modelli semplificano la soluzione del problema aggiungendo due equazioni di trasporto ed introducendo una viscosità turbolenta, denominata Eddy-Viscosity, necessaria al calcolo degli sforzi di Reynolds.

Questi modelli sono adatti a moltissimi campi dell'ingegneria e tipicamente rispondono bene ai livelli di precisione richiesti. Dato che non esiste un modello di turbolenza universale deve essere fatta una scelta ad hoc per ogni differente applicazione.

Di seguito si riportano le descrizioni dei principali modelli di turbolenza con i relativi campi di applicazione.

### 2.4.1 Spalart Allamaras

Si tratta di un modello relativamente semplice ad un'unica equazione che risolve un'equazione di trasporto modellata sulla viscosità cinematica turbolenta.

Il modello di Spalart-Allmaras è stato sviluppato specificamente per il campo aeronautico ed aerospaziale per flussi in parete (intubati) e da buoni risultati per strati limite soggetti a gradienti di pressione avversa. E' abbastanza utilizzato anche nelle applicazioni riguardanti le turbomacchine.

## 2.4.2 Modello k- $\epsilon$

L'approccio k- $\epsilon$  appartiene ai modelli a due equazioni i quali storicamente sono stati largamente utilizzati per descrivere la turbolenza nei flussi industriali.

Risolve due equazioni di trasporto e modella gli sforzi di Reynolds utilizzando l'approccio della viscosità turbolenta. Robustezza, economia computazionale e ragionevole precisione per una vasta gamma di applicazioni industriali ne spiegano la sua popolarità, soprattutto nel campo della trasmissione del calore.

Tuttavia risulta insensibile ai gradienti avversi di pressione ed agli effetti di distacco dello strato limite, risulta quindi inadatto per applicazioni di aerodinamica esterna e nella modellazione di flussi transonici nelle turbomacchine.

## 2.4.3 Modello k- $\omega$

L'equazione di trasporto di  $\omega$  offre diversi vantaggi rispetto alla precedente. Il principale vantaggio consiste nel fatto che l'equazione può essere integrata nel substrato viscoso senza termini aggiuntivi e tipicamente risulta più adatto a descrivere gradienti di pressioni avverse e separazione dello strato limite. Tuttavia l'equazione di trasporto di  $\omega$  rende la soluzione molto sensibile ai valori di k ed  $\omega$  del flusso indisturbato e perciò il suo utilizzo è sconsigliato.

Il modello di turbolenza SST k- $\omega$  è stato sviluppato per ovviare a questo problema combinando elementi dell'equazione  $\omega$  ed  $\epsilon$ . In aggiunta il modello è stato calibrato per calcolare con precisione il distacco dello strato limite dalle superfici, e pertanto

è considerato lo stato dell' arte dei modelli a due equazioni e da risultati più precisi del modello di Spalart-Allamaras.

Nonostante sia possibile utilizzarlo anche a basso numero di Reynolds, in questo caso è preferibile il modello a 4 equazioni SST-Transition essendo in grado di gestire la transizione da regime laminare a turbolento. Questo comporta una maggiore precisione soprattutto per il calcolo del Drag ed è il modello che è stato scelto per l'ottimizzazione.

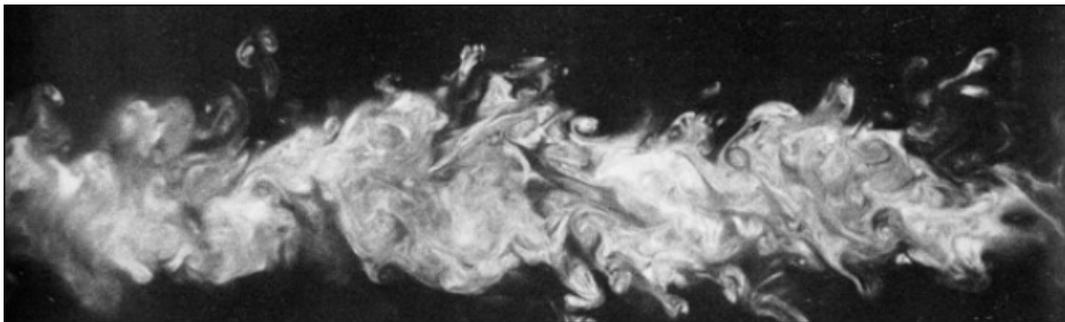


Figura 5 esempio di corrente turbolenta

## 3. Algoritmi Genetici

Un algoritmo genetico (AG) è un algoritmo iterativo che opera su una popolazione di individui che codificano le possibili soluzioni di un dato problema. Gli individui sono valutati tramite una funzione che ne misura la capacità di risolvere il problema e identifica i più adatti alla riproduzione. La nuova popolazione si evolve in base ad operatori random, ispirati alla riproduzione e alla mutazione. Il ciclo completo è ripetuto fino al raggiungimento di un dato criterio di fermata.

### 3.1 Il modello di Holland

Il modello originario di Holland si basa su una popolazione di  $n$  stringhe di bit di lunghezza fissata  $l$ , generate in modo casuale. L'insieme delle stringhe binaria di lunghezza  $l$  ha  $2^l$  elementi e rappresenta lo spazio delle soluzioni del problema. Ogni stringa (genotipo) è la codifica binaria di una soluzione candidata (fenotipo). In generale la funzione di fitness si presenta nella forma seguente:

$$F = f(x_1, x_2, \dots, x_n)$$

Tramite questa funzione, a ogni genotipo  $g_i$  della popolazione iniziale  $P(t=0)$  è associato un valore  $F_i = F(g_i)$  che rappresenta la capacità dell'individuo di risolvere il problema dato. Per determinare il valore di adattività, la funzione di fitness riceve in input un genotipo, lo decodifica nel corrispondente fenotipo e lo testa sul problema dato. Una volta conclusa la fase di valutazione degli individui della popolazione

iniziale, si genera una nuova popolazione  $P(t+1)$  di nuove  $n$  soluzioni candidate applicando gli operatori di selezione, crossover, mutazione e inversione che verranno chiariti nei prossimi paragrafi.

### **3.1.1 Selezione**

All'interno di una popolazione, ad ogni individuo è associata una probabilità di selezione legata alla fitness. L'operatore di selezione genera un numero casuale che determina quale individuo verrà scelto. L'individuo selezionato viene copiato nel cosiddetto mating pool. Il mating pool è così riempito di  $n$  copie degli individui selezionati, al tempo  $P(t=0)$ . La nuova popolazione  $P(t+1)$  è ottenuta attraverso gli operatori di crossover, mutazione e inversione. L'operatore di selezione, scegliendo gli individui che hanno la possibilità di generare discendenti con fitness alta, gioca nel contesto dell'algoritmo genetico il ruolo della selezione naturale per gli organismi viventi.

Esistono diversi metodi per selezionare gli individui tra questi i due più utilizzati sono la "Roulette" ed il "Tournament". Nel primo caso si opera come in una vera e propria roulette dando una maggior probabilità di essere selezionati agli individui che presentano un fitness maggiore.

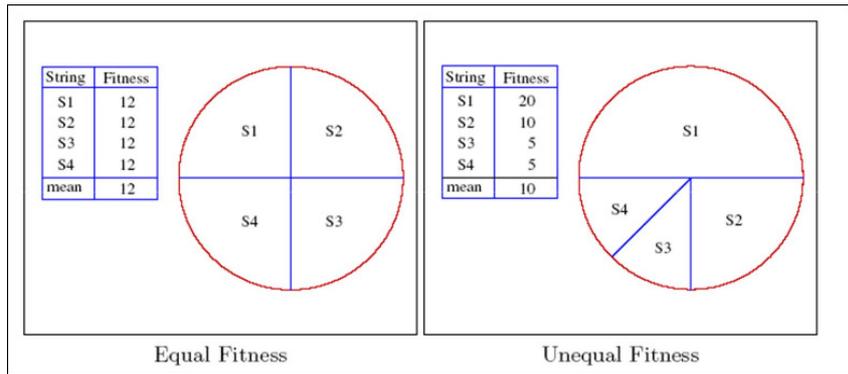


Figura 6 esempio di selezione a roulette

Per quanto riguarda il metodo a torneo invece vengono semplicemente selezionati due individui in maniera casuale e della coppia viene tenuto quello con fitness maggiore.

### 3.1.2 Crossover

All'interno del mating pool sono scelti a caso due individui, detti genitori, e un punto di taglio, detto punto di crossover. Le porzioni di genotipo alla destra del punto di crossover sono scambiate generando due discendenti, come in figura

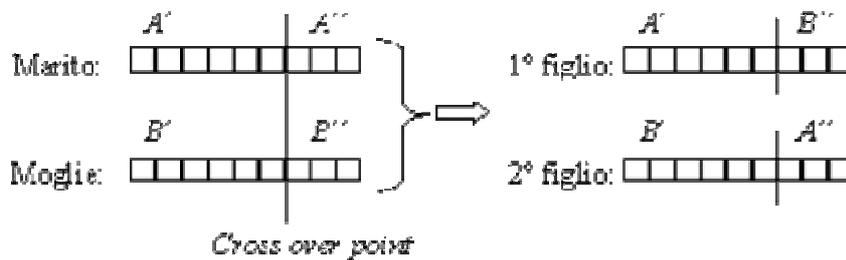


Figura 7 esempio di crossover a singolo punto

Il cosiddetto operatore di crossover one point è applicato  $n/2$  volte per ottenere  $n$  discendenti in base ad una prefissata probabilità  $p$ . Nel caso in cui il crossover non sia applicato, i discendenti coincidono con i genitori.

Un'altra tecnica utilizzata è il two points crossover: gli individui non sono rappresentati da stringhe lineari, ma da cerchi. Quindi si sostituirà una porzione di cerchio di un individuo con quella di un altro selezionando due punti di crossover.

Il crossover uniforme invece prevede che per ogni coppia di genitori si generi una stringa binaria della stessa lunghezza chiamata maschera. Il discendente viene generato copiando il bit del padre o quello della madre a seconda che nella corrispondente posizione della maschera vi sia uno 0 o un 1. Il crossover è una metafora della riproduzione sessuale in cui il materiale genetico dei discendenti è una combinazione di quello dei genitori.

### 3.1.3 Mutazione

Questo operatore è ispirato alla rara variazione di elementi del genoma degli esseri viventi durante l'evoluzione. In base a una probabilità piccola  $p$ , viene cambiato il valore dei bit di ogni individuo (da 0 a 1 e viceversa).



Come in natura, la mutazione aggiunge un "rumore" o una certa casualità all'intera procedura, in modo da assicurare che partendo da una popolazione generata

casualmente non vi siano punti dello spazio delle soluzioni che non vengano esplorati.

### 3.1.4 Inversione

In base a una fissata probabilità  $p$ , sono scelti casualmente due punti nella stringa che codifica l'individuo e inverte i bit tra le due posizioni.



Su una popolazione iniziale numerosa è difficile stimare quali valori della probabilità di crossover e della probabilità di mutazione daranno le migliori performances. L'esperienza mostra che vi è una forte dipendenza dal tipo di problema. Generalmente la probabilità di crossover è tra il 60 e l'80%, mentre quella di mutazione oscilla fra 0,1-1%. Se le probabilità che un individuo sia selezionato per la riproduzione sono proporzionali alla sua fitness (se  $f$  è il valore di fitness di una soluzione e  $F$  la somma dei valori di fitness di tutta la popolazione, la probabilità potrebbe essere  $f/F$ ), è probabile che, in seguito a crossover, gli individui migliori vengano ricombinati, con la conseguente perdita del cromosoma migliore. Per evitare ciò ed accelerare i tempi di convergenza si può clonare l'individuo migliore di una generazione. Attraverso tale tecnica, detta elitismo, mantenendo un alto numero di popolazioni, è possibile clonare più individui nella successiva generazione, mentre per gli altri si procede in modo classico.

## 3.2 Algoritmo genetico multi obiettivo

Una delle caratteristiche più interessanti degli algoritmi genetici è la loro facilità nel trattare problemi di ottimizzazione multi-obiettivo. La soluzione di un problema a più obiettivi è rappresentata da un insieme di punti, detto fronte di pareto, in esso ogni punto descrive una possibile soluzione al problema di pari "dignità" con le altre appartenenti al fronte. Per comprendere meglio il concetto di fronte di pareto è necessario chiarire un altro termine fondamentale, dominanza delle soluzioni. Una soluzione è dominata da un'altra quando tutti gli obiettivi del problema di quest'ultima hanno un fitness migliore. La figura seguente mostra un esempio di fronte in un problema bi-obiettivo, si noti come le soluzioni si addensino su un vero e proprio fronte.

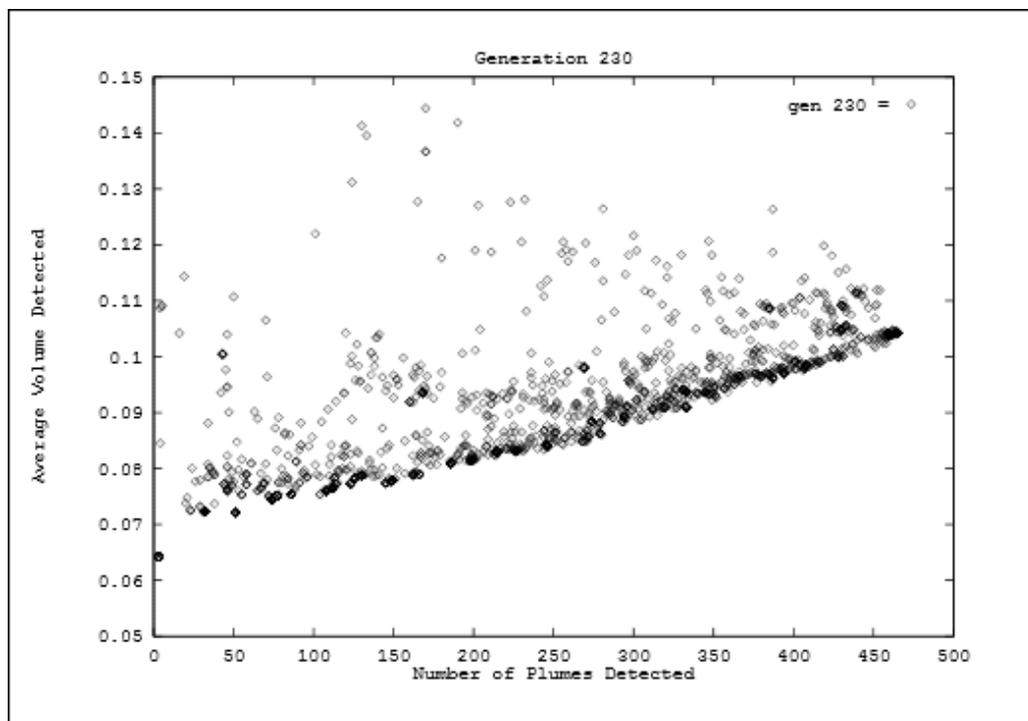


Figura 8 esempio di fronte di pareto



Tale metodo viene impiegato per aumentare la convergenza dell' algoritmo verso il fronte di pareto. Esistono altri algoritmi per l' ordinamento di pareto ma non verranno trattati in questa sede. Nella figura seguente si presentalo schema di un possibile un possibile algoritmo genetico multi-obiettivo.

- **Algoritmo multi-obiettivo:**
  - t=0
  - Inizializza la popolazione dei genitori in modo random
  - **While** t=t\_max
    - Valutazione delle funzioni obiettivo
    - Ordinamento di Pareto (Pareto Ranking)
    - Selezione (diverse possibilità)
    - Ricombinazione
    - Mutazione
    - (eventuale ulteriore meccanismo di preservazione della diversità genetica)
    - (eventuale elitarismo)
    - Nuova popolazione
    - t=t+1
  - **End**

### 3.3 Metodi di preservazione della diversità genetica

All' aumentare delle generazioni e quindi con il procedere dell' ottimizzazione il materiale genetico tende ad impoverirsi, questo può portare ad un fronte di pareto non completo ed ad un insufficiente esplorazione del dominio. Per ovviare al problema sono stati implementati nell' algoritmo genetico i metodi di preservazione della diversità genetica, ne citiamo i più importanti:

- Fitness sharing
- Crowding

- Accoppiamento ristretto

La filosofia comune a tutti questi metodi è quella di penalizzare l' accoppiamento tra individui vicini tra loro mediante l' utilizzo di nicchie opportunamente definite.

### 3.3.1 Fitness sharing

Nel fitness sharing la nicchia è una sfera (3 dimensioni) o un ipersfera (in più dimensioni). Il raggio di questa sfera è un parametro definito dall' utente e tipicamente viene fatto decrescere col progredire dell' ottimizzazione, la scelta del valore iniziale è quindi arbitraria e questo risulta essere il suo limite.

Una volta definita la nicchia viene contato il numero di individui che le appartengono assegnando una penalità al fitness delle ipersfere più numerose. Di fatto è una funzione di penalità mascherata che ha lo scopo di aiutare la mutazione a sparpagliare gli individui lungo tutto il fronte di pareto .

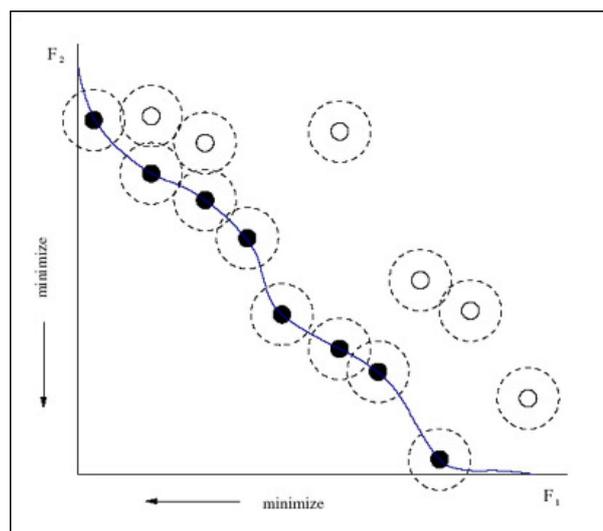


Figura 10 nicchie circolari per il fitness sharing bi-obiettivo

### 3.3.2 Crowding

Si basa sulla stessa filosofia del fitness sharing, ossia privilegia le zone vuote, ma con due principali differenze. La prima è che in questo caso la nicchia è data da un iperprisma e la seconda ad essa correlata è il fatto che le dimensioni del prisma non sono arbitrarie, infatti esso è generato a partire da due individui selezionati sulla base di un criterio che tiene conto del grado di affollamento nello spazio degli obiettivi. Un altro fattore importante risiede nel fatto che tale metodo non penalizza solo le soluzioni appartenenti al prisma ma anche quelle ad esso vicine.

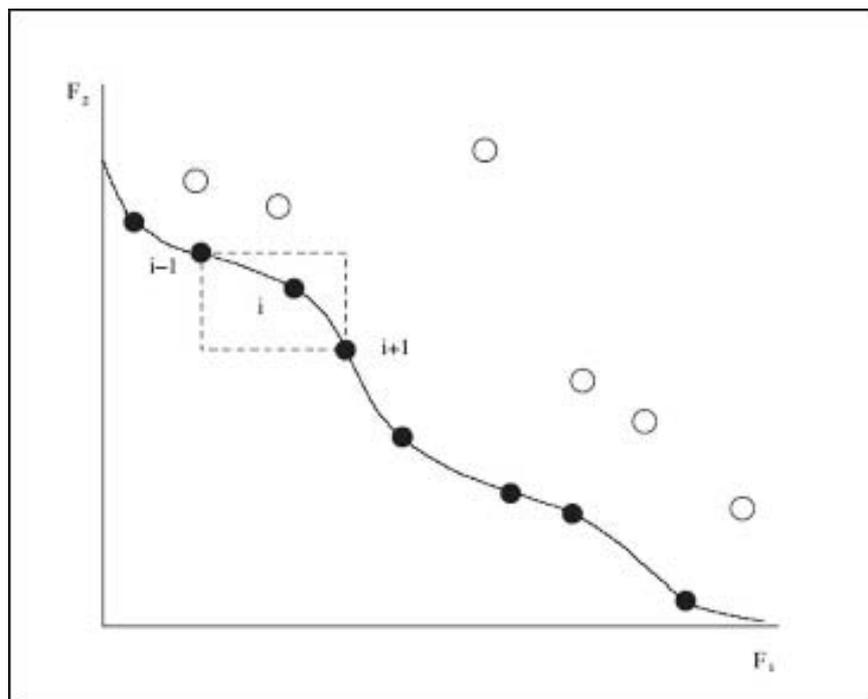


Figura 11 esempio di crowding

### **3.3.3 Accoppiamento ristretto**

Nell' accoppiamento ristretto l' algoritmo impedisce agli individui di accoppiarsi nel caso la loro distanza nello spazio degli obbiettivi si eccessivamente bassa.

## 4. Parametrizzazione del profilo Clark-y

Essendo il Profilo di partenza, mutuato dal database della NASA, composto di 52 punti per intradosso ed estradosso, una delle priorità è dovuta essere necessariamente la parametrizzazione della geometria al fine di diminuire le variabili in gioco.

Si è optato per una parametrizzazione tramite curve di Bezier che, oltre a ridurre sensibilmente il numero delle variabili, assicurano una deformazione della geometria regolare ed uniforme. Questo è dovuto al fatto che la deformazione dovuta allo spostamento di uno dei punti di appoggio del poligono di Bezier interessa una zona piuttosto estesa del profilo.

Di seguito si riporta la forma parametrica di tali curve:

$$B(t) = \sum_{i=0}^n \binom{n}{i} P_i (1-t)^{n-i} t^i = P_0 (1-t)^n + \binom{n}{1} P_1 (1-t)^{n-1} t + \dots + P_n t^n, \quad t \in [0, 1].$$

Dove i polinomi:

$$\binom{n}{i} (1-t)^{n-i} t^i$$

Sono detti polinomi di Bernstein.

Essendo la formula parametrica di Bezier non invertibile, si è creata una funzione ad hoc per poter replicare la geometria sfruttando l' algoritmo genetico di MATLAB, tale funzione minimizza la somma delle distanze tra i punti della geometria di

partenza e la quelli calcolati secondo la formula di Bezier. La funzione PARAM.m, presente in appendice, può essere descritta sinteticamente nei seguenti punti:

- Importazione dei punti del profilo dal database
- Assegnazione dei punti fissi del poligono di Bezier (estremi)
- Calcolo della curva tramite la funzione di Bezier
- Calcolo della funzione di errore tra punti reali e di Bezier
- Creazione di un File contenente i punti di controllo ed il valore della funzione di fitness (nel caso in questione l' errore)

Si sono così ottenuti 7 punti di controllo sia per l' estradosso che per l' intradosso:

xbi	0	0	123.7300	600	498.0100	927.7500	1000
ybi	0	-45.3500	-48.3700	0	-26.1400	-1.6100	-0.6000
xbe	0	0	123.7300	600	498.0100	927.7500	1000
ybe	0	35.3000	199.6600	13.9900	146.5900	13.1600	0.6000

Dai punti di controllo di estradosso ed intradosso si sono poi ricavati i corrispondenti punti della linea media e del semi-spessore tramite le formule:

```
importBEZ('infracontrol.txt');
xbi=data(1,:);
ybi=data(2,:);
importBEZ('extracontrol.txt');
xbe=data(1,:);
ybe=data(2,:);

xlm=(xbi+xbe)/2;
ylm=(ybi+ybe)/2;
xs=xlm;
ys=ybe-ylm;
```

Al fine di evitare l'incrocio tra intradosso ed estradosso il profilo è stato parametrizzato attraverso la sua linea media alla quale i semi-spessori vengono aggiunti per ricavare l'estradosso e sottratti per intradosso, in questo modo è possibile anche far variare indipendentemente sia il lato in pressione sia quello in depressione.

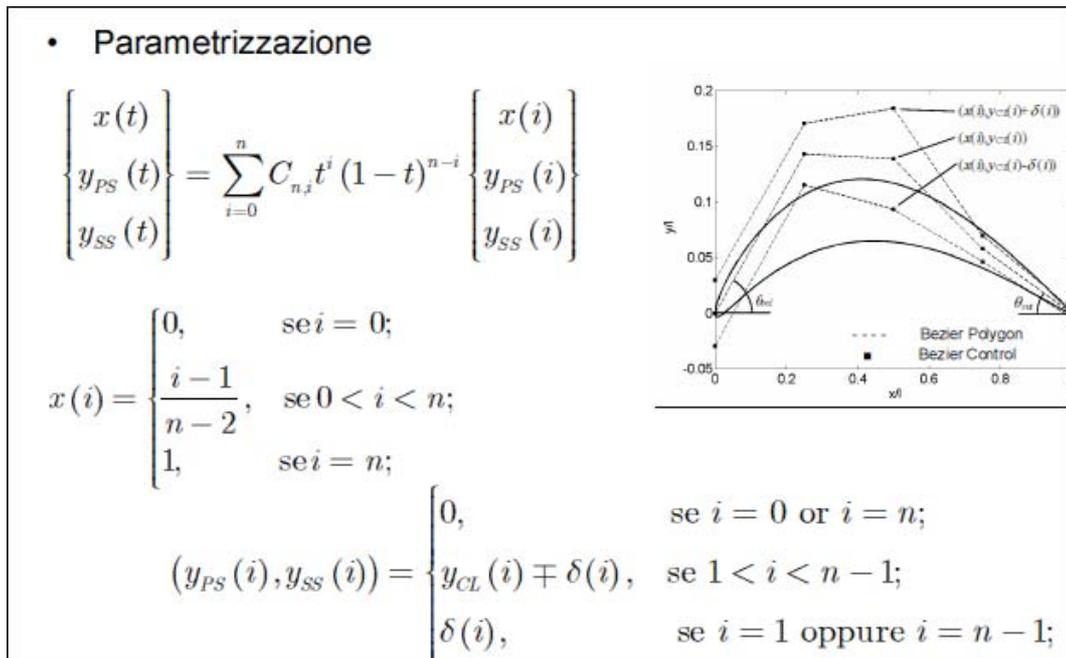


Figura 12 esempio di parametrizzazione tramite curve di Bezier

## 4.1 Creazione del dominio di calcolo

Il software utilizzato per creare la griglia di calcolo è GAMBIT il quale tramite un file di journal genera un file con estensione .msh che viene letto da FLUENT il solutore fluidodinamico.

Tra le numerose tipologie di mesh è stata scelta una mesh strutturata a celle rettangolari perché, sebbene sia meno flessibile rispetto ad una mesh ibrida ad elementi triangolari con boundary layer rettangolare, risulta più efficiente e quindi più adatta a gestire l' elevato numero di simulazioni richieste per un'ottimizzazione tramite algoritmo genetico.

Il processo di creazione del dominio di calcolo può essere riassunto nei seguenti punti:

- Creazione di un file contenente le coordinate dei punti
- Avvio del programma
- Importazione dei punti
- Definizione dei confini
- Creazione della mesh
- Esportazione

Definiti i punti di controllo di bezier la geometria del profilo è stata calcolata tramite la seguente funzione di matlab:

```
xpe=0;
ype=0;
t=linspace(0,1,ee); %vettorizzazione di t
for i=0:n-1
    c=factorial(n-1)/(factorial(i)*(factorial(n-1-i)));
    xpe=xpe+c.*t.^i.*(1-t).^(n-1-i).*xbe(i+1); %con grandezze
    vettoriali *=.*
    ype=ype+c.*t.^i.*(1-t).^(n-1-i).*ybe(i+1);
end
xpe=xpe';
ype=ype';
```

ed i punti del profilo sono stati salvati in un file di testo opportunamente formattato per poter essere letto da gambit, se ne riporta un esempio.

0.00000	0.00000	0.00000
0.00046	-0.13589	0.00000
0.00186	-0.27146	0.00000
0.00419	-0.40672	0.00000
0.00746	-0.54166	0.00000
0.01167	-0.67628	0.00000

La prima colonna rappresenta le coordinate x del profilo, quella centrale le coordinate y ed infine le coordinate z (nulle nel nostro caso essendo bidimensionale). Come si può notare è necessario che non ci siano spazi tra le righe mentre deve esserci almeno uno spazio vuoto tra le colonne. Le coordinate inoltre devono essere ordinate dal naso alla coda del profilo.

Il file journal di Gambit viene creato in automatico mentre si esegue manualmente la procedura di seguito descritta.

## 4.2 Importazione dei punti del profilo

La geometria del profilo viene importata in Gambit tramite il comando **File** →

**Import** → **vertex data**

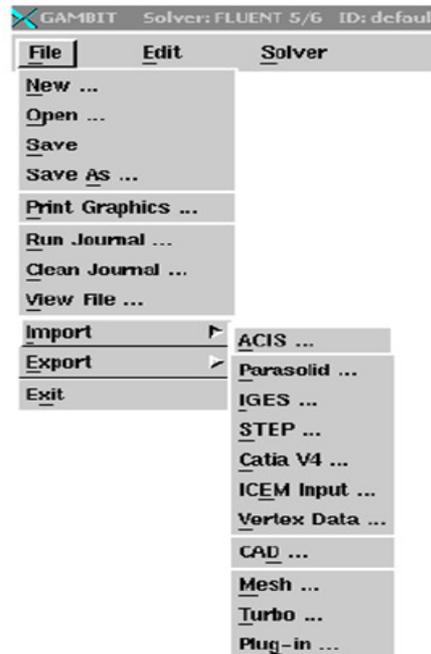


Figura 13 importazione dei punti in Gambit

Dopo aver selezionato il file di testo contenente i punti del profilo, i vertici creati sono stati interpolati con delle spine cubiche in modo da ottenere i due "edge" assegnati ad intradosso ed estradosso .Il profilo poi è stato virtualmente diviso ad un terzo della corda tramite il comando **split edge**. Il contorno della mesh è formato da un semicerchio e due rettangoli tale tipologia di mesh è definita C-mesh e risulta essere una delle più diffuse per simulazioni di profili alari. I punti di

appoggio per la mesh esterna sono stati creati tramite il comando **Create real Vertex** in **Geometry Operations**.

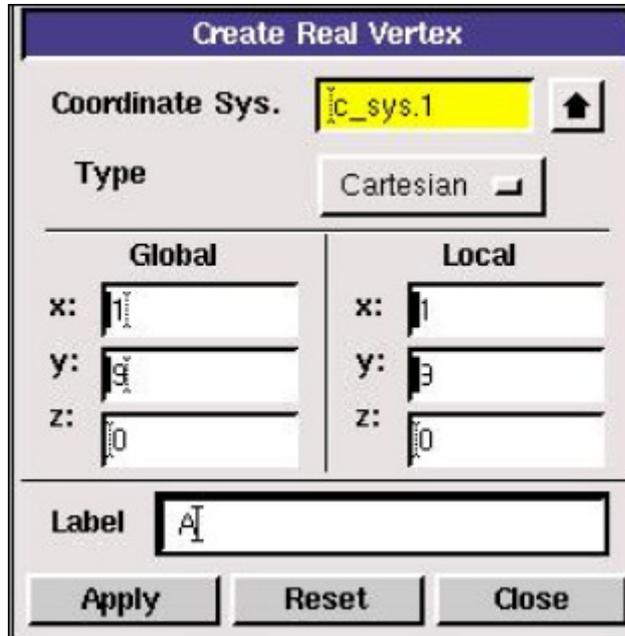


Figura 14 Creazione dei punti esterni del dominio

Utilizzando il sistema di coordinate globali è possibile creare i punti di appoggio per la geometria di confine attorno al profilo. I valori nella tabella seguente rappresentano gli estremi del contorno esterno della mesh sarà quindi definito da 9 vertici.

Label	x-coordinate	y-coordinate	z-coordinate
A	c	9*c	0
B	14.4*c	9*c	0
C	14.4*c	UT	0
D	14.4*c	0	0
E	14.4*c	LT	0
F	14.4	-9*c	0
G	C	-9*c	0
H	C	0	0
I	-8*c	0	0

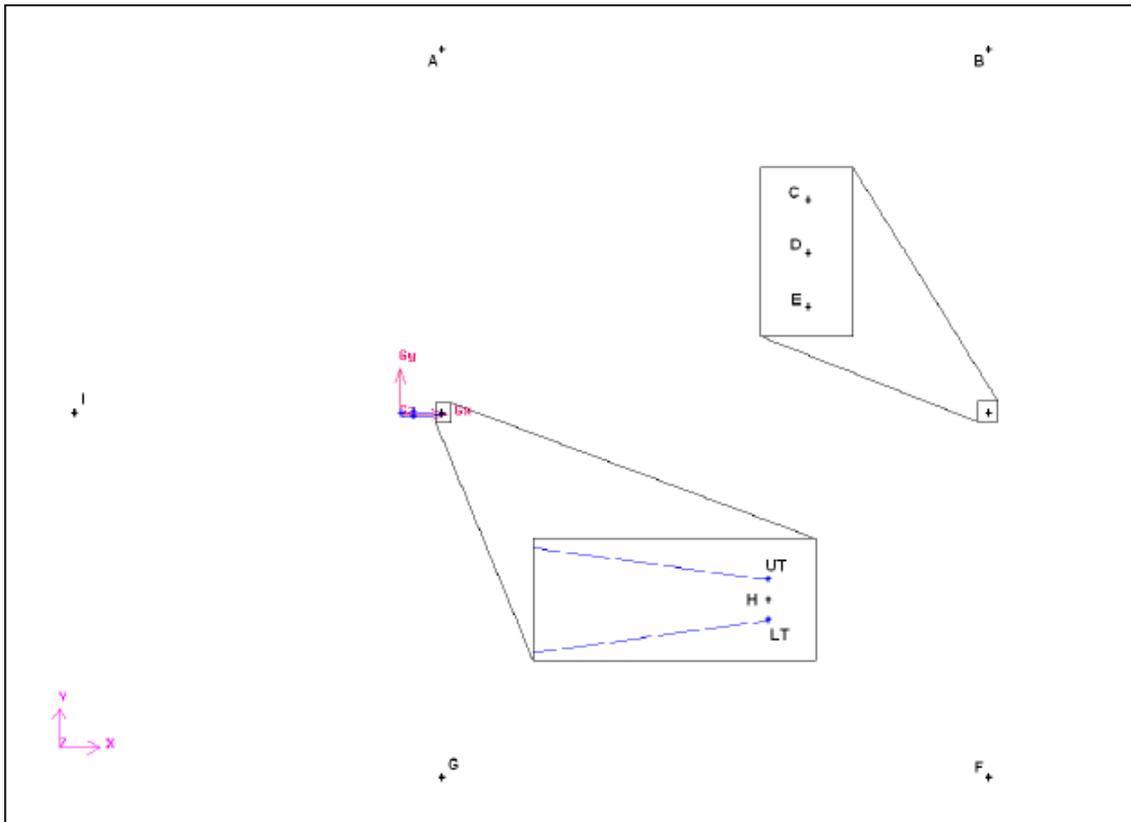


Figura 15 Particolari del dominio di calcolo

I valori UT ed LT corrispondono ,come mostra la figura, ai valori della coordinata y per estradosso ed intradosso, tale " gap" risulta importante per la creazione di una mesh tridimensionale e si è quindi ritenuto opportuno tenerlo in previsione di un futuro sviluppo del lavoro.

Il termine C in tabella si riferisce alla corda e nel nostro caso ha valore unitario.

I vertici I,G,A sono stati uniti tramite una semicirconferenza mentre tutti gli altri sono stati collegati con segmenti rettilinei.

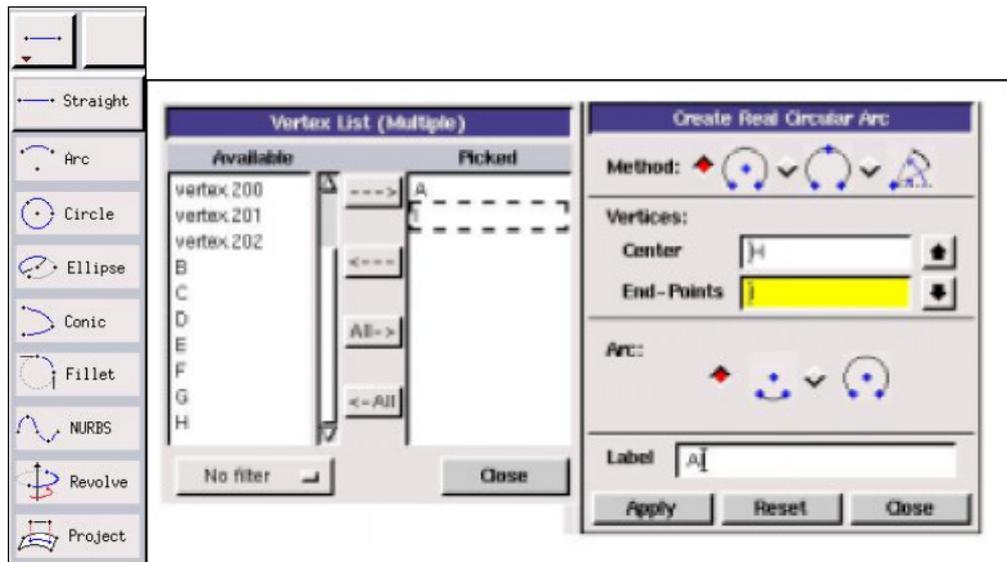


Figura 16 Generazione della faccia esterna

Le superfici così create sono state sottratte tra di loro tramite un'operazione booleana in modo da creare un'unica faccia che presenta un foro delle dimensioni del profilo al centro.

### 4.3 Mesh 2D

La regione più prossima al profilo è quella con il campo di moto di maggior interesse, è quindi necessario che il calcolo nelle zone adiacenti ad essa sia il più preciso possibile e, di conseguenza, che la griglia abbia densità maggiore in prossimità di esso. Il primo step per la meshatura del dominio è la suddivisione degli spigoli che la compongono, ciò è stato realizzato utilizzando il comando **mesh edge** controllando la dimensione degli elementi con il comando **first length** che permette di decidere la dimensione della prima cella dello spigolo e riadatta

automaticamente le successive in funzione della lunghezza dello spigolo e del numero di punti in cui è suddiviso.

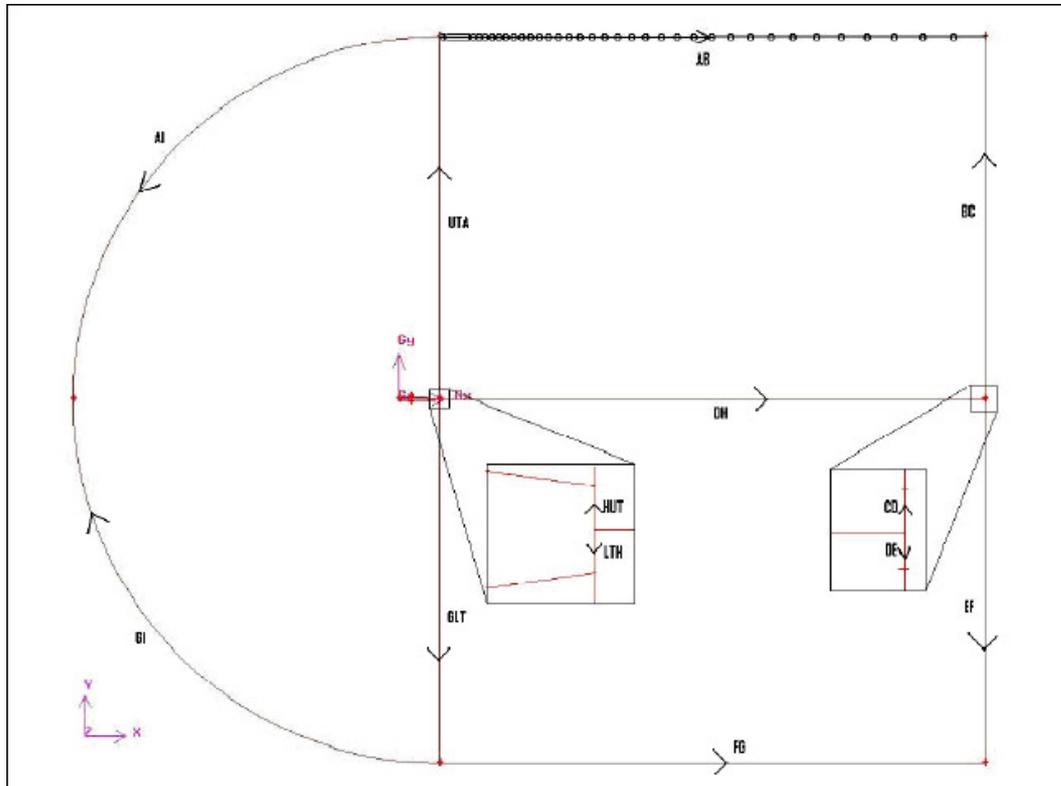


Figura 17 Suddivisione degli "edge" della mesh

Estradosso ed intradosso sono stati suddivisi in 170 elementi, 60 per il primo terzo della corda e 110 per la restante parte infittendo la mesh vicino a naso e coda, dove le prime celle hanno una lunghezza pari a 0.005 mm. Lo schema utilizzato per la mesh è il **Map** a celle rettangolari il quale richiede che il numero di elementi sia conforme per ogni faccia, di conseguenza il semicerchio è stato suddiviso in 120 elementi. Di fondamentale importanza è stata la scelta dell' altezza della prima cella per la discretizzazione della regione che contiene lo strato limite, infatti da essa dipende il valore di  $Y^+$ . Il valore raccomandato in letteratura per  $Y^+$  è nell' intervallo

aperto ]0,1[ si è dunque proceduto per step successivi a determinare l' altezza della prima cella che rendesse idoneo tale valore. E' stata scelta come altezza 0.005 alla quale corrisponde nelle condizioni operative un valore di  $Y^+$  pari a 0.4 e quindi nella norma. Il risultato finale della procedura ha generato una mesh di 159000 elementi.

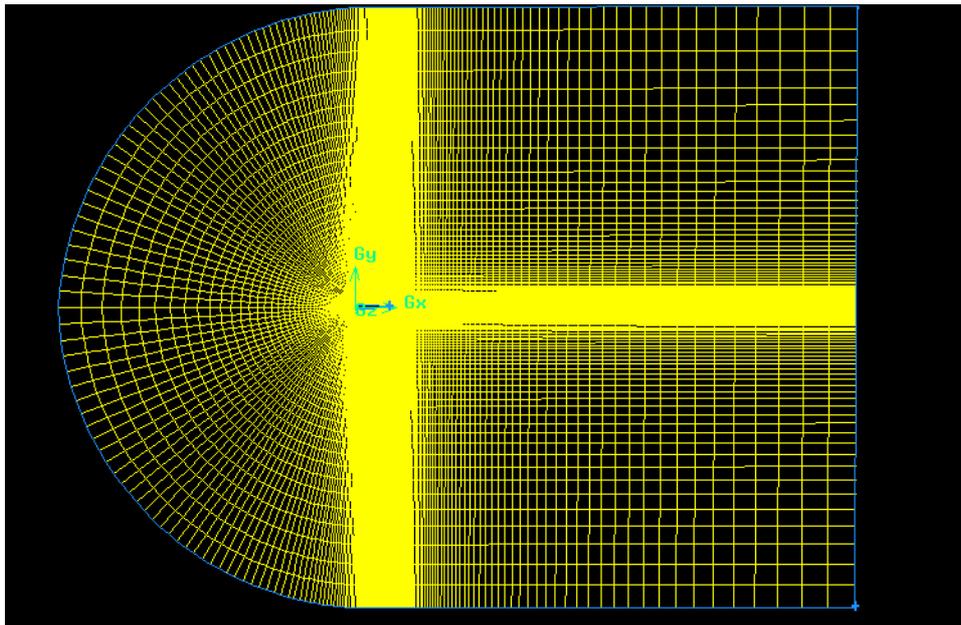
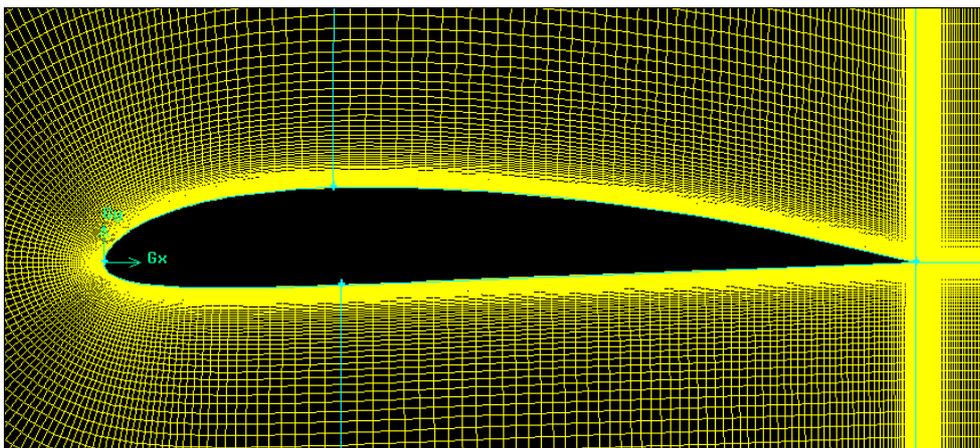


Figura 18 Griglia di calcolo: vista completa e di dettaglio



Come si può notare dalle immagini precedenti la distribuzione dei nodi è tale da rendere le celle il più possibile ortogonali al profilo in modo da ridurre lo "skewing"

della griglia. Alti valori di questo parametro influenzano negativamente le performance della simulazione rendendola instabile e talvolta non convergente. L'elemento peggiore della mesh ha un valore di **equisizeskew** pari a 0.18 il che rende la mesh molto performante essendo estremamente basso. Particolare attenzione è stata posta anche nel relegare gli elementi ad elevato **aspect ratio** lontani dal profilo in modo da evitare errori di approssimazione nel calcolo delle grandezze da parte del solutore.

Per quanto riguarda la zona adibita a contenere la scia sono stati effettuati particolari accorgimenti in termini di aspect-ratio assicurandosi che le celle adiacenti al trailing edge fossero quadrate.

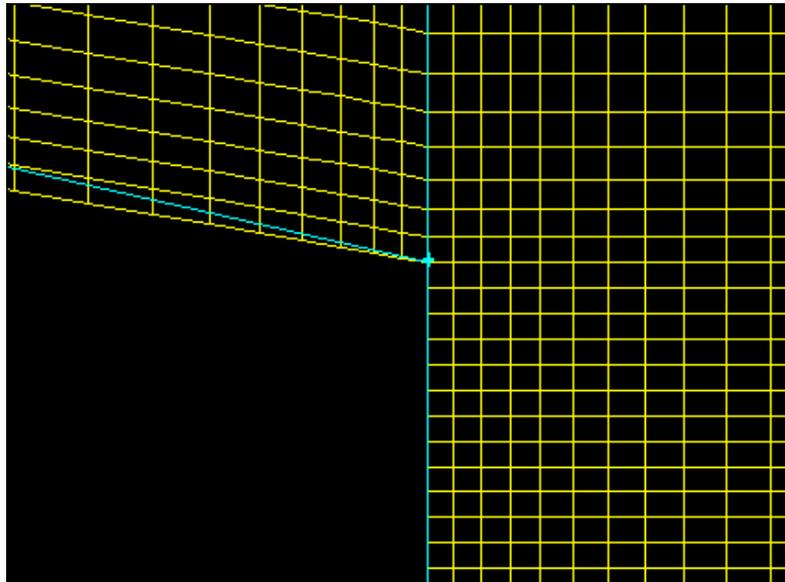


Figura 19 Griglia di calcolo: dettaglio della coda del profilo

## 5. Validazione del modello fisico

In questo capitolo si riportano le curve caratteristiche sperimentali del profilo di partenza confrontandole con quelle numeriche e le condizioni al contorno imposte nel solutore per poterle ricavare.

Le curve sperimentali sono state mutuate da un report di M.S. Rice e sono riportate nella seguente figura dove sono stati sovrapposti in rosso i punti derivanti dalla simulazione

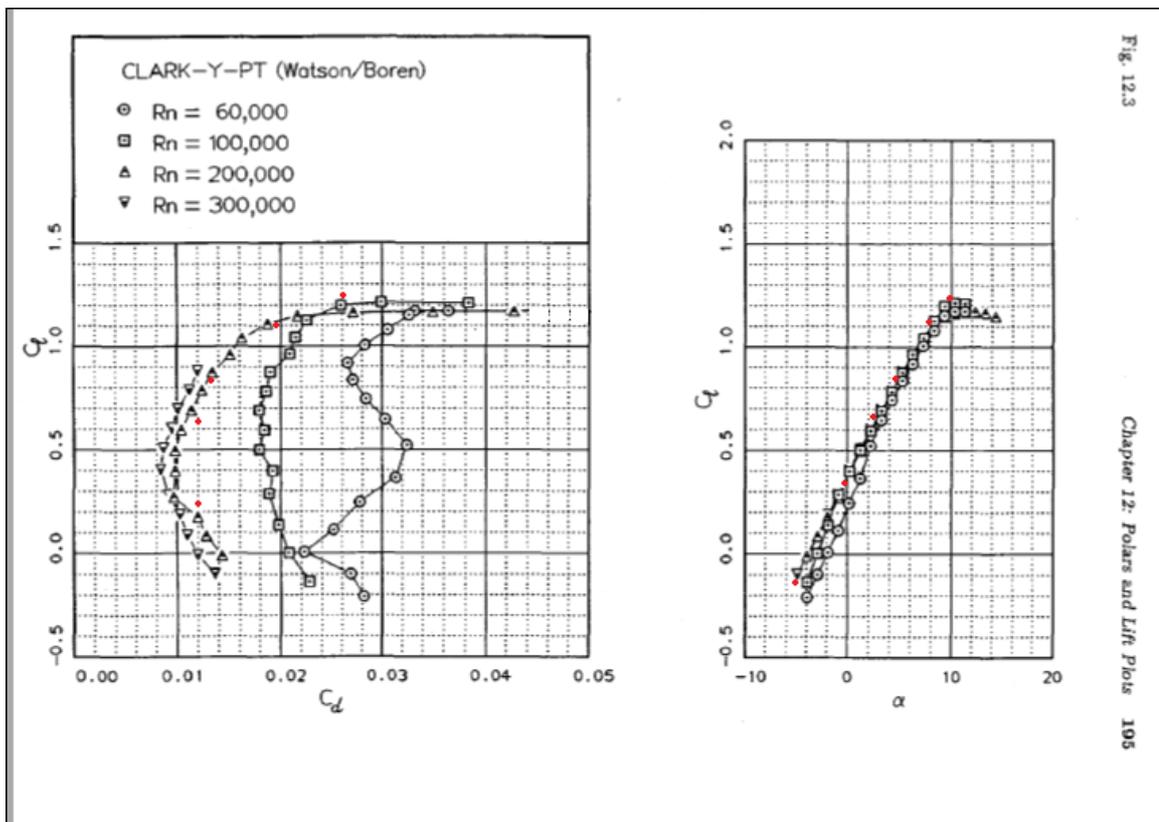


Fig. 12.3

Figura 20 Confronto tra le polari sperimentali e quelle numeriche

Per poter lavorare in similitudine è stato necessario nel corso della validazione riportarsi nelle condizioni operative della prova sperimentale, e questo è assicurato dal numero di Reynolds :

$$Re = \frac{\rho v l}{\mu}$$

Dove le grandezze al denominatore sono rispettivamente densità del fluido, velocità della corrente e lunghezza della corda, mentre al denominatore compare la viscosità dinamica. Il numero di Reynolds misura il rapporto tra le forze di inerzia e quelle di natura viscosa, è quindi adimensionale e descrive bene il fenomeno fisico del passaggio da moto laminare a moto turbolento.

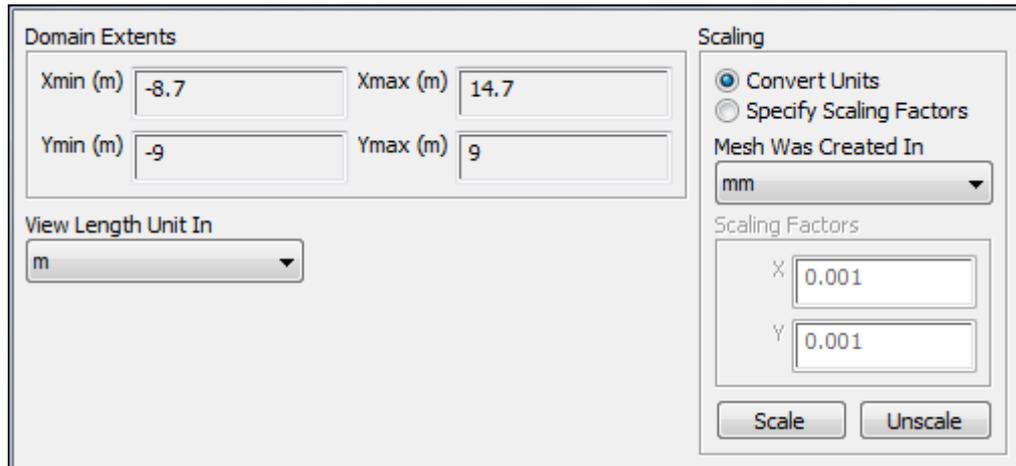
Avendo il nostro profilo corda unitaria la condizione di similitudine meccanica si verifica per una velocità della corrente fluida pari 5.3685 m/s che corrisponde ad un valore del numero di Reynolds pari a 300000.

### 5.3 Modello numerico

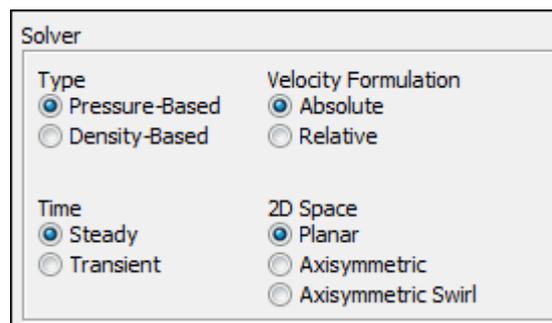
La geometria creata tramite Gambit è stata salvata con estensione **.msh** , tale formato è adatto ad essere importato in Fluent al fine di simulare il profilo.

Di seguito vengono riportata la procedura seguita per simulare il profilo, la quale è stata poi tradotta in un file journal che provvede a replicarla in maniera automatica.

La griglia di calcolo, creata in millimetri, è stata scalata in modo da avere corda unitaria, quindi ogni coordinata è stato moltiplicata per un fattore 0.001.

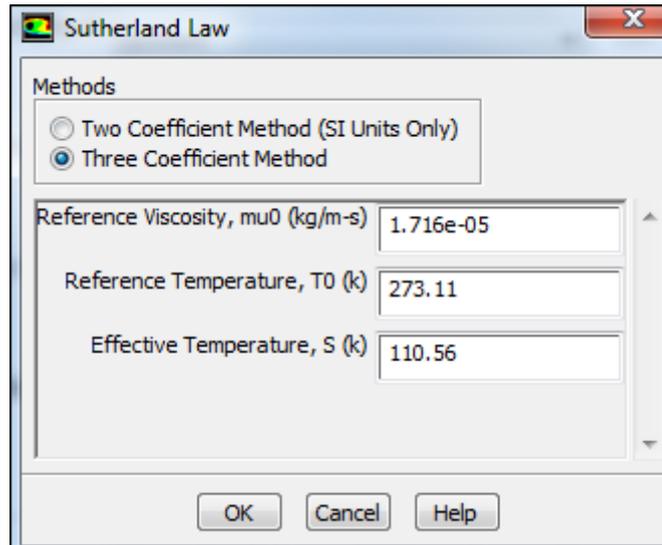


Il solutore è stato impostato come **pressure-based** in modalità stazionaria e quindi indipendente dal tempo, per la velocità si è mantenuta una formulazione assoluta.



Sono stati successivamente abilitati il modello di turbolenza (kw-SST) e l'equazione dell'energia necessaria per un calcolo della densità più preciso.

Si è passato poi alla definizione delle caratteristiche del fluido operativo che risulta essere aria standard per la quale la densità, grazie all'equazione dell'energia, viene calcolata secondo la legge dei gas ideali. La viscosità è invece calcolata secondo lo schema di Sutherland a tre coefficienti.



Il valore della pressione operativa, preimpostata a 101325 Pa, è stata lasciato inalterato e si è dunque proceduto a definire le condizioni al contorno del dominio di calcolo. L'intero semicerchio e i due lunghi segmenti orizzontali ai confini della mesh sono stati impostati come **velocity-inlet**, e quindi all'ingresso è stata imposta una velocità costante in modulo e direzione. La direzione della velocità, definita dai coseni direttori dell'angolo di attacco, è stata fatta variare di volta in volta e sono stati necessari quindi un journal dedicato per ogni valore dell'incidenza. Il segmento verticale, all'estrema destra del dominio, è stato impostato come **pressure-outlet** e quindi soggetto a pressione costante data dalle condizioni operative del fluido.

Lo schema risolutivo delle RANS è stato impostato come **P-V-Coupled** e quindi le equazioni di pressione e velocità sono risolte contemporaneamente in modalità pseudo-transitoria, questo ha permesso di raggiungere la convergenza con un numero di iterazioni relativamente basso (130).

Per quanto riguarda la discretizzazione spaziale delle grandezze lo schema del terzo ordine è risultato più robusto in termini di convergenza e stabilità delle simulazioni, oltre ad essere più preciso e leggermente più oneroso a livello computazionale. Pressione e gradiente sono stati discretizzati con gli schemi del secondo ordine non essendo disponibili nel solutore quelli del terzo.

Per il calcolo della temperatura totale è stata utilizzata l'espressione della gasdinamica in funzione del numero di Mach di seguito riportata:

$$T^o = T \left( 1 + \frac{\gamma - 1}{2} M^2 \right)$$

Dove  $\gamma$  è il rapporto tra i calori specifici , 1.4 per l'aria, ed M il numero di Mach.

Per quantificare il grado di convergenza e la stabilità dei risultati si è deciso di monitorare i residui( con criterio di arresto ad  $1 \times 10^{-6}$  )ed i grafici del lift e del drag.

La soluzione è stata inizializzata in modalità ibrida la quale risolve l'equazione di Laplace per produrre campi di velocità conformi a geometrie dal dominio complesso.



## 6. Ottimizzazione

Gli obiettivi posti per questo progetto sono quelli di aumentare l'efficienza aerodinamica dell'ala andando a determinare per quale angolo d'attacco essa sia massima senza penalizzarla nelle altre configurazioni.

Si è deciso dunque di formulare un problema bi-obiettivo cercando di massimizzare la funzione di fitness:

$$f = cl * \left(\frac{cl}{cd}\right)^2$$

Tale funzione viene calcolata per un range di angoli compresi tra [-5,10] e salvata in un vettore contenente sia l'angolo che il valore della funzione.

Il secondo obiettivo, al fine di ottenere una curva di efficienza il più piatta possibile, è stato formulato come la massimizzazione dell'area sottesa dalla curva  $f(\alpha)$ . Il calcolo di tale area è stato ottenuto implementando il metodo dei trapezi direttamente nella funzione obiettivo.

La procedura di ottimizzazione è stata automatizzata in modo da essere il più efficiente possibile. Il programma deputato a controllare le attività durante lo svolgimento è MATLAB al quale è deputato il compito di aprire gambit e fluent facendo leggere ad essi gli opportuni file di journal, a leggere i risultati del simulatore, a calcolare le funzioni obiettivo ed a far girare, tramite il toolbox di ottimizzazione l'algoritmo genetico.

Il cuore dell' ottimizzazione risiede infatti proprio nel file KITGEN.m presente in appendice e brevemente descritto nel prossimo paragrafo.

## 6.1 La funzione KITGEN.m

La "function" KITGEN.m accetta un vettore di 15 variabili indipendenti, generate dall' algoritmo genetico ,che corrispondono alle ordinate ed alle scisse dei punti di controllo dei poligoni di Bezier necessari a definire la geometria del profilo e restituisce un vettore a due dimensioni contenenti i valori delle funzioni da massimizzare.

E' importante notare che l' algoritmo presente nel toolbox di matlab può solo minimizzare le funzioni, è stato dunque necessaria porre un segno meno davanti alle funzioni in modo che il minimo della funzione col segno cambiato corrispondesse al massimo della funzione desiderata.

Il vettore in ingresso rappresenta i punti di controllo della linea media del profilo e della curva dei semi-spessori, e tramite un opportuno cambio di coordinate da questi vengono ricavati i punti per l' estradosso e l' intradosso del profilo tramite la funzione:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               CALCOLO GEOMETRIA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xpi=0; %inizializzazione variabile
ypi=0;
t=linspace(0,1,2001); %vettorizzazione di t
for i=0:n-1
    c=factorial(n-1)/(factorial(i)*(factorial(n-1-i)));
    xpi=xpi+c.*t.^i.*(1-t).^(n-1-i).*xbi(i+1);      %con grandezze
vettoriali *=.*
    ypi=ypi+c.*t.^i.*(1-t).^(n-1-i).*ybi(i+1);

```

```

end
xpe=0; %inizializzazione variabile
ype=0;
t=linspace(0,1,2001); %vettorizzazione di t
for i=0:n-1
    c=factorial(n-1)/(factorial(i)*(factorial(n-1-i)));
    xpe=xpe+c.*t.^i.*(1-t).^(n-1-i).*xbe(i+1); %con grandezze
vettoriali *=. *
    ype=ype+c.*t.^i.*(1-t).^(n-1-i).*ybe(i+1);

end
xpi=xpi'; %traspone il vettore riga in colonna
ypi=ypi';
xpe=xpe'; %traspone il vettore riga in colonna
ype=ype';

```

Definita la geometria vengono creati i file di testo contenenti i punti del profilo nell' opportuno formato, precedentemente descritto, necessario per essere letto da Gambit che viene fatto partire in automatico tramite il comando:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                APERTURA GAMBIT E LETTURA JOURNAL DI GAMBIT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dos('C:\Users\Andrea\Documents\MATLAB\TESI\log_gambit.bat');
    fid1 = fopen('giustissima.msh');
    while fid1 < 0
        pause (5);
    end
fclose(fid1);

```

Nel file di batch log\_gambit.bat sono fornite al programma le istruzioni per l' apertura e la lettura del journal di Gambit. Il ciclo while è necessario al fine di attendere che il file con estensione .msh venga creato, in questo modo infatti il programma attende fino alla sua apertura evitando di bloccare l' intera procedura con errori dovuti alla mancata presenza del file.



L' ultima attività svolta dalla funzione KITGEN.m è la creazione di due file, uno (Backup.txt) contenente tutti i punti di controllo di tutti i profili con il loro fitness, e l' altro (Pareto.txt) contenente solo i profili che effettivamente migliorano le prestazioni rispetto ai precedenti in modo da rendere più facile la determinazione dei profili ottimali.

Questa procedura è ripetuta per ogni profilo di generazione in generazione, e le simulazioni dei vari angoli sfruttano ognuno un journal dedicato.

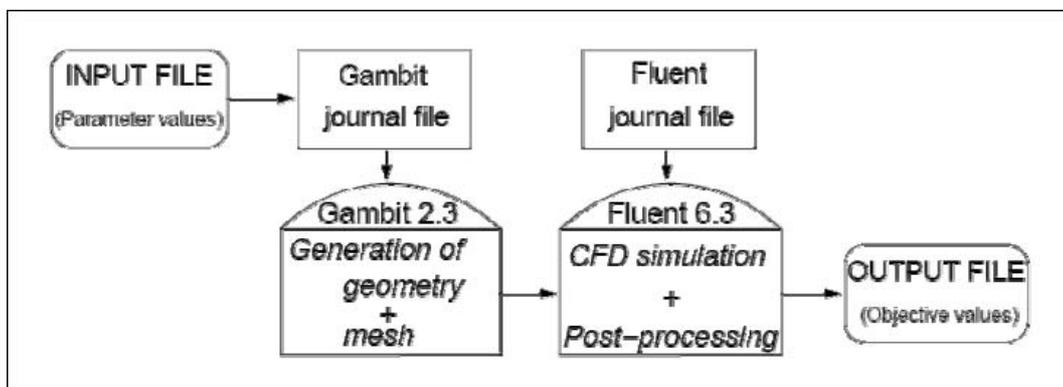


Figura 21 schema a blocchi della procedura di ottimizzazione

## 6.2 Impostazioni dell' algoritmo Genetico

Come già detto l' algoritmo genetico utilizzato è quello presente nel toolbox di ottimizzazione globale di Matlab, il quale permette di decidere i diversi parametri della procedura quali, ad esempio, numero di individui e numero di generazioni, metodo di selezione ,vincoli e modalità di crossover.

Visto il grande numero di simulazioni necessarie per ogni individuo si è scelto di limitare la numerosità della popolazione a 20 individui e sono state simulate 10

generazioni. In prima battuta sono state create 5 generazioni partendo da una popolazione iniziale random, le successive 5 invece sono partite dagli individui migliori precedentemente generati.

Per evitare che i punti generati in maniera random dessero luogo a profili non simulabili sono stati imposti dei vincoli laterali sulle variabili e sono stati individuati per tentativi in modo da limitare l'escursione delle variabili senza impedire all'algoritmo di generare configurazioni potenzialmente utili.

Il metodo di selezione è stato impostato su "tournament" ossia a torneo, dove gli individui selezionati a coppie competono sulla base del fitness ed il migliore dei due viene selezionato.

L'"elite count" è stato impostato a 2 ossia i 2 migliori individui di ogni popolazione vengono riportati automaticamente nella successiva. La scelta di questo parametro è basata sul fatto che aumentando il numero di individui "elite" si rischia di avere generazioni "incestuose" e quindi di limitare lo spazio di ricerca dell'algoritmo.

La modalità di crossover è stata impostata su "scattered" ossia crossover uniforme descritto nel capitolo 3 riguardante gli algoritmi genetici.

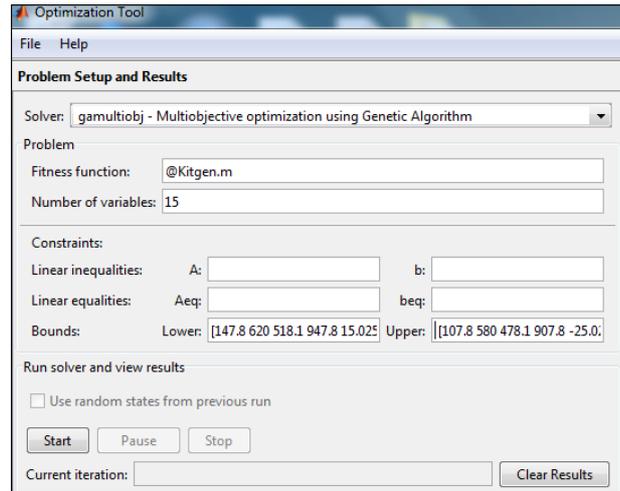
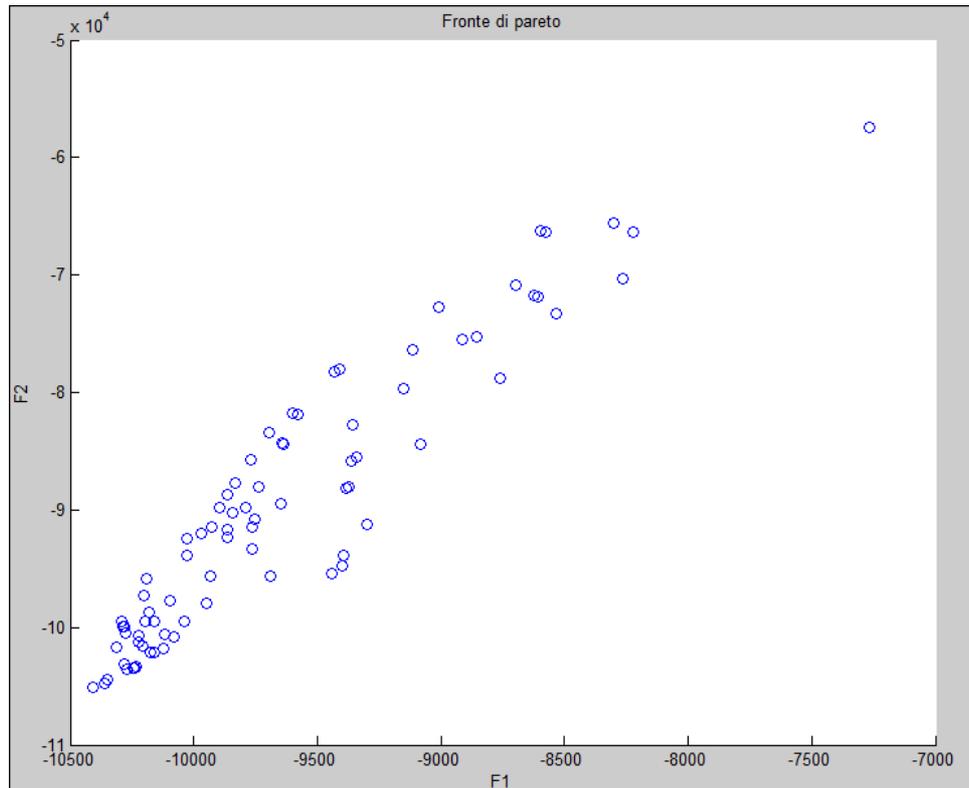


Figura 22 Schermata di lancio dell' algoritmo genetico multi obiettivo

## 6.3 Fronte di pareto



Lo scopo dell'algorithm genetico è quello di andare a trovare il fronte di Pareto, cercando di infittire il numero di individui su di esso, rendendoli il più omogeneamente possibile distribuiti.

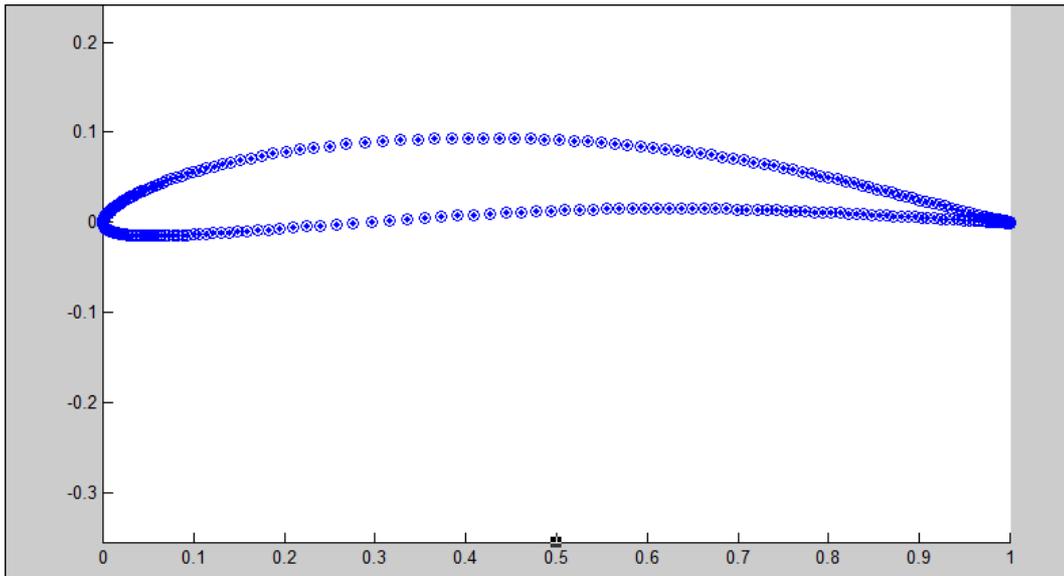
In tal modo si potrà scegliere l'individuo appartenente al fronte più utile alla nostra applicazione, non avendo di per se un problema multi-obiettivo un'unica soluzione.

Essendo i due obiettivi strettamente correlati fra loro l'algorithm non ha potuto generare un vero e proprio fronte perché un miglioramento nella prima funzione di fitness comporta una miglioria nella seconda funzione.

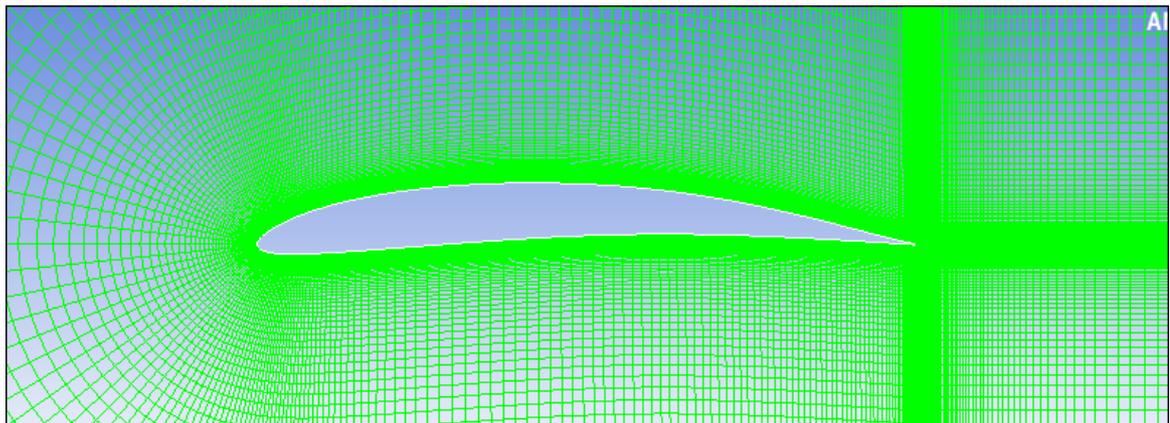
Nonostante il fronte non si sia sviluppato l'ottimizzazione ha comunque prodotto un profilo aerodinamico migliore di quello di partenza le cui caratteristi verranno descritte nel prossimo paragrafo. Si fa notare in oltre che il fronte riportato è quello di un problema minimo-minimo in quanto come già detto in precedenza l'algorithm genetico di matlab minimizza le funzioni di fitness, che nel nostro caso (massimo-massimo) sono state cambiate di segno in modo da ovviare al problema.

## **6.4 Soluzione ottimale**

Si riportano i dati della soluzione candidata come ottimale la cui geometria è rappresentata nelle figure seguenti nella vista per punti :



e come dominio di calcolo:



I punti di controllo di Bezier che identificano il profilo sono:

$XBE=[0 \ 0.07 \ 127.8 \ 600 \ 498.1 \ 927.8 \ 1000]$

$YBE=[0 \ 15.75612 \ 180.14003 \ 30.17134 \ 180.05438 \ 20.63996 \ 0.6]$

XBI=[0 0.07 127.8 600 498.1 927.8 1000]

YBI=[0 -47.80612 22.17333 2.78901 60.24197 5.16849 -0.6]

Si nota subito che il profilo ottimale rispetto a quello di partenza ha un camber maggiore ed un spessore minore nel prossimo paragrafo verrà descritto come questi cambiamenti influenzano le prestazioni.

L'efficienza massima del profilo si presenta per un angolo di attacco di 5 gradi ed ha un valore di 94.4 nettamente superiore a quella del profilo di partenza.

## 6.5 Confronto

In questo capitolo si presentano e commentano i grafici relativi alle prestazioni dei profili, in particolare vengono confrontati i coefficienti di lift e drag l'efficienza ed il coefficiente di pressione.

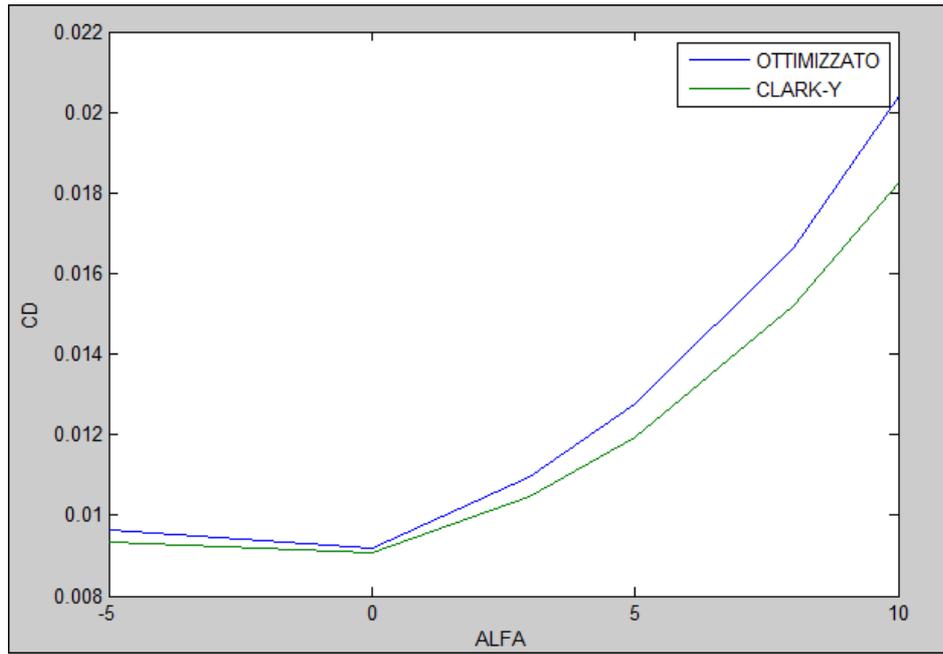


Figura 23: Coefficiente di Drag, confronto

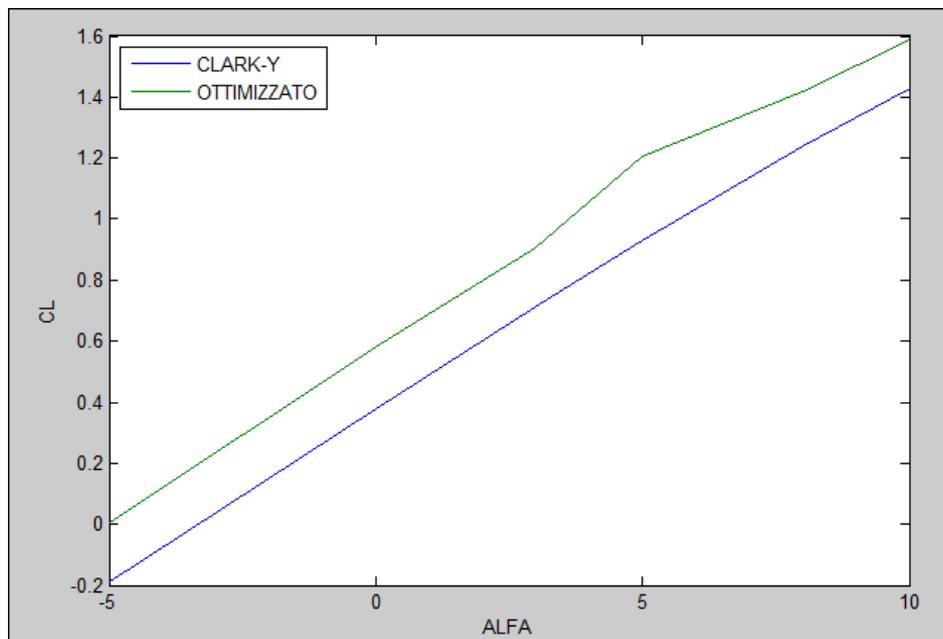


Figura 24: Coefficiente di Lift, confronto

Come ci si aspettava, la maggior curvatura del profilo comporta un peggioramento del coefficiente di drag che risulta lievemente superiore per il profilo ottimizzato. Questo effetto è compensato in parte da uno spessore minore che determina una resistenza aerodinamica, a parità di curvatura, più bassa. A fronte della maggior resistenza aerodinamica, il camber maggiorato del profilo ottimizzato determina un sensibile aumento del coefficiente di lift producendo complessivamente un effetto positivo sull'efficienza aerodinamica del profilo. È da notare inoltre che la soluzione scelta risulta più stabile oltre che più efficiente andando ad eliminare la deportanza per angoli di attacco negativi.

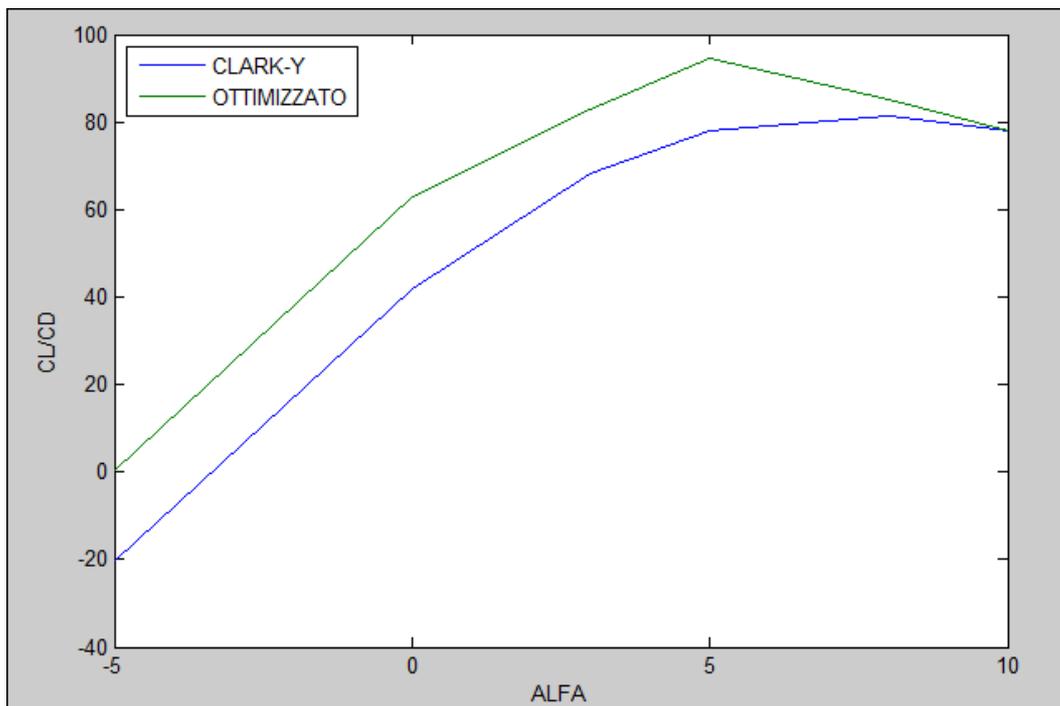


Figura 25: Efficienza aerodinamica, confronto

L'andamento del coefficiente di pressione, riportato nella figura seguente, testimonia e quantifica la maggior efficienza del profilo ottimizzato, si nota infatti come l'area compresa tra le curve del lato in pressione e depressione sia maggiore per la soluzione adottata. Degno di nota anche il minor gradiente sul lato in depressione che determina una maggiore stabilità della corrente fluida.

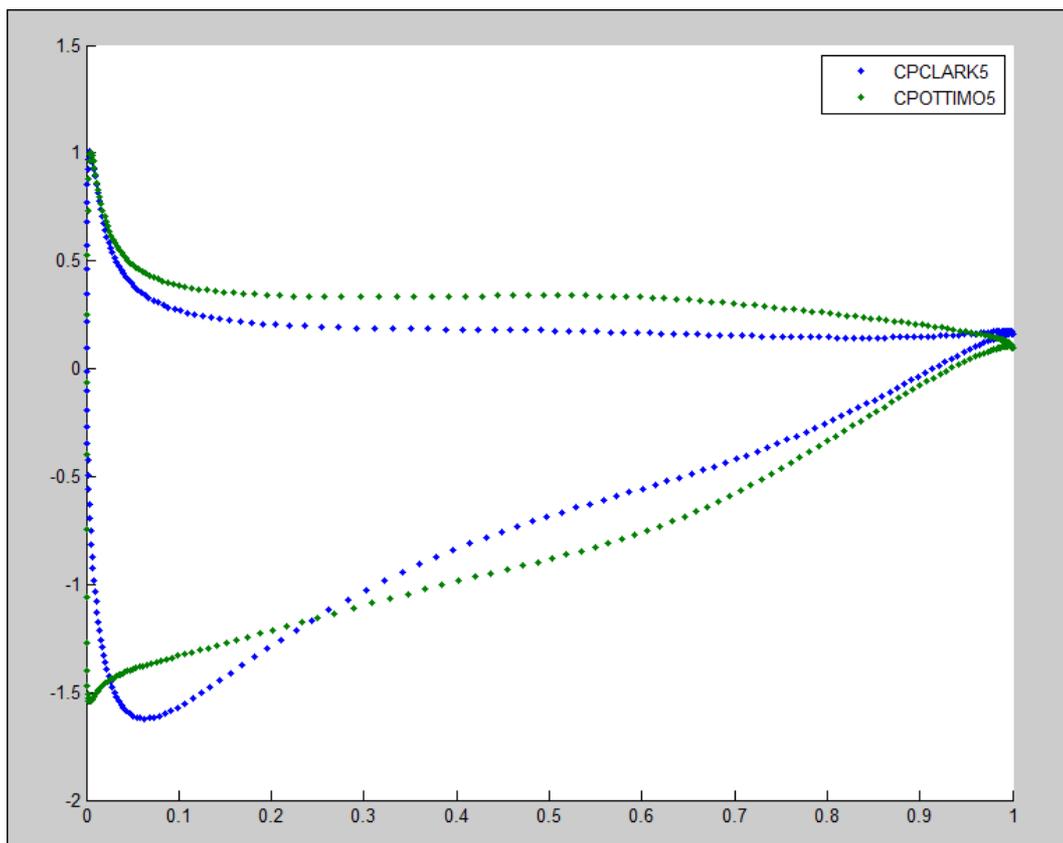


Figura 26: Coefficiente di pressione, confronto

Nonostante il coefficiente di drag sia leggermente superiore per il profilo ottimizzato la variazione può considerarsi minima e, come ci si aspettava, le scie dei due profili lo mettono in evidenza essendo molto simili a livello di spessore dello strato limite e per quanto riguarda i valori di pressione totale.

Le figure che seguono mettono in evidenza la scia dei due profili in due condizioni operative differenti: rispettivamente per gli angoli di attacco 5 e 10 gradi. Si può notare facilmente la crescita dello strato limite nella configurazione più inclinata in accordo con l' aumento del coefficiente di drag.

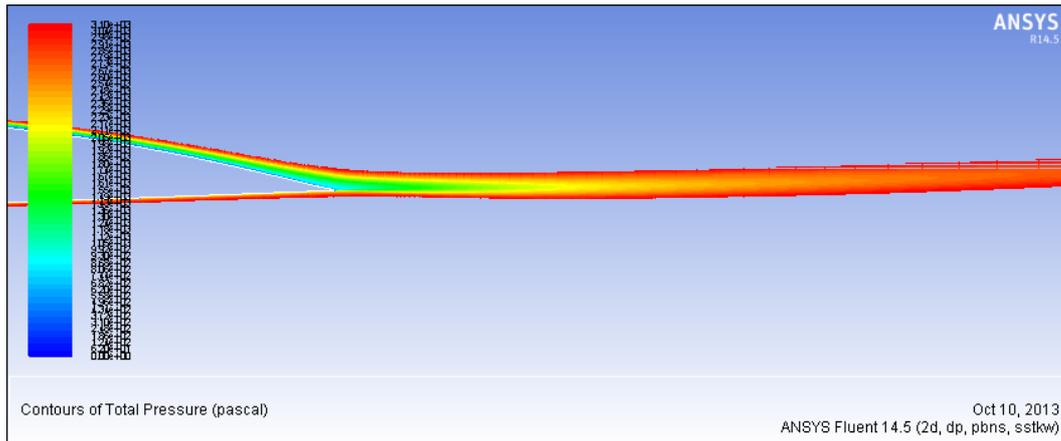


Figura 27: Scia del profilo clark-y, angolo di attacco 5°

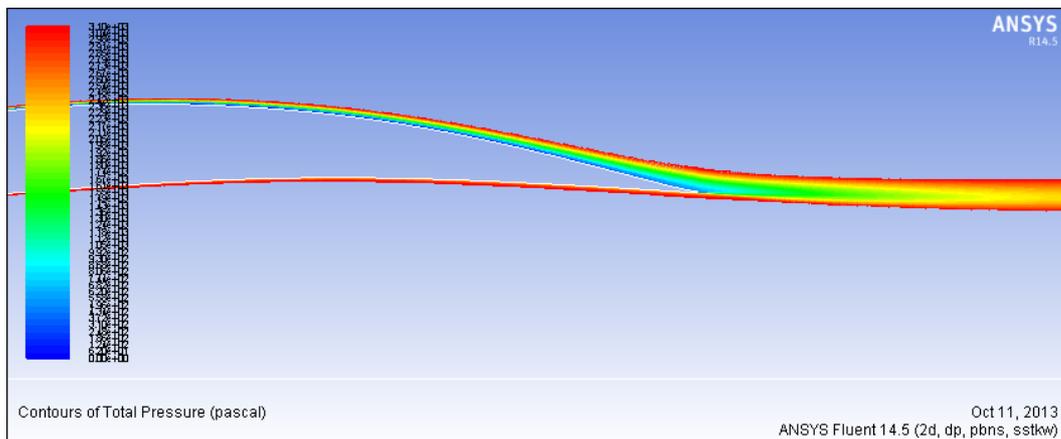


Figura 28: Scia del profilo ottimizzato, angolo di attacco 5°

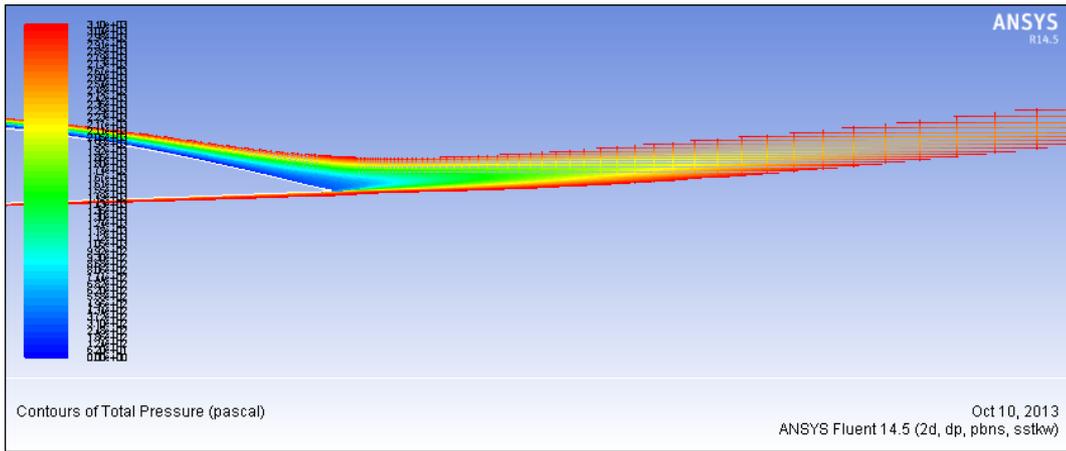


Figura 29: Scia del profilo clark-y, angolo di attacco 10°

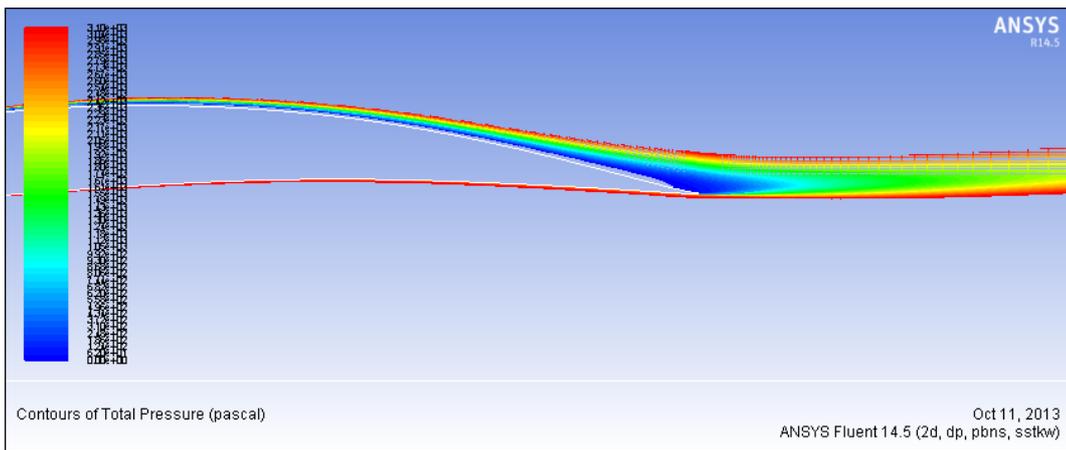


Figura 30: Scia del profilo ottimizzato, angolo di attacco 10°



## 7. Conclusioni e sviluppi

Dall' analisi dei risultati esposti in precedenza si è messo in evidenza come il profilo ottimizzato sia superiore per tutti gli angoli di attacco simulati sia in termini di efficienza sia in termini di portanza. Dal confronto del coefficiente di pressione la soluzione designata sia soggetta ad una forza risultante maggiore, essendo rappresentata dall' area compresa tra le curve del  $C_p$  tra intradosso ed estradosso. Nel corso dell' ottimizzazione questi risultati sono stati ottenuti dall' algoritmo genetico aumentando il camber del profilo e riducendone lo spessore. L' effetto di tali modifiche è stato un aumento dell' efficienza per tutti gli angoli di attacco simulati ed una maggiore stabilità dovuta ad un gradiente di pressione minore sul lato in depressione del profilo. In particolare è stato risolto il problema del profilo Clark-Y per angoli di incidenza negativi: dove quest' ultimo presenta un coefficiente di lift negativo la soluzione adottata ha portanza leggermente positiva.

La scelta di simulare un intervallo limitato di angoli di attacco è stata necessaria per ovviare ai fenomeni non stazionari della corrente per incidenze inferiori a  $-5^\circ$  e superiori a  $10^\circ$ , si ritiene interessante lo sviluppo del lavoro in campo tridimensionale con capo di moto non stazionario( e quindi con simulazioni di tipo transient). Non è stato possibile lo studio dei fenomeni non stazionari in campo tridimensionale per questioni legate alla tempistica essendo necessaria una tipologia di simulazione onerosa dal punto di vista computazionale. Il lavoro si presta bene per lo sviluppo di un modello numerico con interazione fluido

struttura, dove oltre a valutare le grandezze fluidodinamiche si può descrivere il loro effetto in termini di deformazioni dei materiali costruttivi del kite.



```

xbe(n)=xe(ee);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               ORDINAMENTO DELLE VARIABILI DECISIONE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:n-2
    xbe(i+1)=xey(i);

    ybe(i+1)=xey(i+(n-2));

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               CALCOLO ESTRADOSSO BEZIER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xpe=0;
ype=0;
t=linspace(0,1,ee); %vettorizzazione di t
for i=0:n-1
    c=factorial(n-1)/(factorial(i)*(factorial(n-1-i)));
    xpe=xpe+c.*t.^i.*(1-t).^(n-1-i).*xbe(i+1); %con grandezze
    vettoriali *.=.*
    ype=ype+c.*t.^i.*(1-t).^(n-1-i).*ybe(i+1);
end
xpe=xpe';
ype=ype';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               CALCOLO FITNESS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=xpe-xe;
B=ype-ye;

fitness=(sum(sqrt((A).^2+(B).^2)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               TEST FITNESS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fid=fopen('infracontrolkite.txt','rt');
TEST=fscanf(fid,'FITNESS %f');
fclose(fid);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               GRAFICI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

plot(xpe,ype,xe,ye)
if fitness<TEST
fid=fopen('infracontrolkite.txt','at');

```

```

    fprintf(fid,'FITNESS %3.5f \r\n',...
            fitness);
    fprintf(fid,'xbe %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f
%3.5f %3.5f \r\n',...
            xbe);
    fprintf(fid,'ybe %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f
%3.5f %3.5f \r\n',...
            ybe);

    fclose(fid);
end

return

```

## Appendice B: KITGEN.m

```

function [fitness]=KITEGEN(xey)
% xey è un vettore di 3n-6 elementi
% xey=xm3,xm4,xm5,xm6,ym2,ym3,ym4,ym5,ym6,ys2,ys3,ys4,ys5,ys6
% xbe ed xbi sono i vettori dei punti di controllo di estradosso ed
% intradosso
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               IMPOSTAZIONI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=7; %numero punti di controllo di ogni curva di bezier
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               PREALLOCAZIONE SPAZIO VARIABILI DECISIONE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xm=zeros(n,1);
ym=zeros(n,1);
ys=zeros(n,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               CAMBIO COORDINATE DA LM+S A ESTRADOSSO INFRADOSSO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:n-3

    xm(i+2)=xey(i);

end

xm(1)=0.00000;
xm(2)=0.07;
xm(7)=1000.00;

```

```

for i=5:9
    ym(i-3)=xey(i);
    ys(i-3)=xey(i+5);
end
ym(1)=0.00000;
ym(7)=0.00000;
ys(1)=0.00000;
ys(7)=0.6;

xbe=xm;
ybe=ym+ys;
xbi=xbe;
ybi=ym-ys;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                                CALCOLO GEOMETRIA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xpi=0; %inizializzazione variabile
ypi=0;
t=linspace(0,1,2001); %vettorizzazione di t
for i=0:n-1
    c=factorial(n-1)/(factorial(i)*(factorial(n-1-i)));
    xpi=xpi+c.*t.^i.*(1-t).^(n-1-i).*xbi(i+1);      %con grandezze
vettoriali * = .*
    ypi=ypi+c.*t.^i.*(1-t).^(n-1-i).*ybi(i+1);

end
xpe=0; %inizializzazione variabile
ype=0;
t=linspace(0,1,2001); %vettorizzazione di t
for i=0:n-1
    c=factorial(n-1)/(factorial(i)*(factorial(n-1-i)));
    xpe=xpe+c.*t.^i.*(1-t).^(n-1-i).*xbe(i+1);      %con grandezze
vettoriali * = .*
    ype=ype+c.*t.^i.*(1-t).^(n-1-i).*ybe(i+1);

end
xpi=xpi'; %traspone il vettore riga in colonna
ypi=ypi';
xpe=xpe'; %traspone il vettore riga in colonna
ype=ype';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                                GENERAZIONE FILE PER JOURNAL DI GAMBIT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Z_zeroe=zeros(size(xpe));
Z_zeroi=zeros(size(xpi));
punti_estradosso=[xpe,ype,Z_zeroe];
punti_infradosso=[xpi,ypi,Z_zeroi];
punti_estradosso=punti_estradosso';
punti_infradosso=punti_infradosso';

fid=fopen('estradossopergambit.n','wt');

fprintf(fid,'%8.5f\t %8.5f\t %8.5f\n',...
        punti_estradosso);

```

```

fclose(fid);
fid=fopen('infradossovergambit.n','wt');

    fprintf(fid,'%8.5f\t %8.5f\t %8.5f\n',...
            punti_infradosso);
fclose(fid);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               APERTURA GAMBIT E LETTURA JOURNAL DI GAMBIT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dos('C:\Users\Andrea\Documents\MATLAB\TESI\log_gambit.bat');
    fid1 = fopen('giustissima.msh');
    while fid1 < 0
        pause (5);
    end
fclose(fid1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               APERTURA FLUENT E LETTURA JOURNAL DI FLUENT
%
%               ANGOLO DI ATTACCO 10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dos('C:\Users\Andrea\Documents\MATLAB\TESI\fluent_log10.bat');
    fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_10');
    while fid1 < 0
        pause (5);
        fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_10');
    end
fclose(fid1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               LEGGI RISULTATI FLUENT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cd_10');
cd_10=data(6);
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cl_10');
cl_10=data(6);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               CANCELLA FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cd_10')
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cl_10')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               APERTURA FLUENT E LETTURA JOURNAL DI FLUENT
%
%               ANGOLO DI ATTACCO 8
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dos('C:\Users\Andrea\Documents\MATLAB\TESI\fluent_log8.bat');

```

```

fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_8');
while fid1 < 0
    pause (5);
    fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_8');
end
fclose(fid1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%                               LEGGI RISULTATI FLUENT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cd_8');
cd_8=data(6);
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cl_8');
cl_8=data(6);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%                               CANCELLA FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cd_8')
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cl_8')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%                               APERTURA FLUENT E LETTURA JOURNAL DI FLUENT
%
%                               ANGOLO DI ATTACCO 5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

dos('C:\Users\Andrea\Documents\MATLAB\TESI\fluent_log5.bat');
fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_5');
while fid1 < 0
    pause (5);
    fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_5');
end
fclose(fid1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%                               LEGGI RISULTATI FLUENT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cd_5');
cd_5=data(6);
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cl_5');
cl_5=data(6);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%                               CANCELLA FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cd_5')
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cl_5')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%                               APERTURA FLUENT E LETTURA JOURNAL DI FLUENT
%
%                               ANGOLO DI ATTACCO 3

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dos('C:\Users\Andrea\Documents\MATLAB\TESI\fluent_log3.bat');
fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_3');
while fid1 < 0
    pause(5);
    fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_3');
end
fclose(fid1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               LEGGI RISULTATI FLUENT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cd_3');
cd_3=data(6);
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cl_3');
cl_3=data(6);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               CANCELLA FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cd_3')
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cl_3')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               APERTURA FLUENT E LETTURA JOURNAL DI FLUENT
%
%                               ANGOLO DI ATTACCO 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dos('C:\Users\Andrea\Documents\MATLAB\TESI\fluent_log0.bat');
fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_0');
while fid1 < 0
    pause(5);
    fid1 = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_0');
end
fclose(fid1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               LEGGI RISULTATI FLUENT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cd_0');
cd_0=data(6);
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cl_0');
cl_0=data(6);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               CANCELLA FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cd_0')
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cl_0')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%               APERTURA FLUENT E LETTURA JOURNAL DI FLUENT
%               ANGOLO DI ATTACCO -5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

dos('C:\Users\Andrea\Documents\MATLAB\TESI\fluent_log-5.bat');
fidl = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_-5');
while fidl < 0
    pause (5);
    fidl = fopen('C:\Users\Andrea\Documents\MATLAB\TESI\cl_-5');
end
fclose(fidl);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%               LEGGI RISULTATI FLUENT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cd_-5');
cd_meno5=data(6);
importcd('C:\Users\Andrea\Documents\MATLAB\TESI\cl_-5');
cl_meno5=data(6);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%               CANCELLA FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cd_-5')
delete('C:\Users\Andrea\Documents\MATLAB\TESI\cl_-5')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

%               CREAZIONE VETTORI CL CD MERITO

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

CL(6)=cl_10;
CL(5)=cl_8;
CL(4)=cl_5;
CL(3)=cl_3;
CL(2)=cl_0;
CL(1)=cl_meno5;

CD(6)=cd_10;
CD(5)=cd_8;
CD(4)=cd_5;
CD(3)=cd_3;
CD(2)=cd_0;
CD(1)=cd_meno5;

MERITO=zeros(6,1);
ALFA=[-5 0 3 5 8 10];
for i=1:6

```

```

MERITO(i)=CL(i)*(CL(i)/CD(i))^2;

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               GRAFICO CD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[F, In]=max(MERITO);
fitness(1)=-F;
ANGOLO=ALFA(In);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               METODO DEI TRAPEZI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

z=trapz(ALFA,MERITO);
fitness(2)=-z;
grid on; axis equal
plot(xpe,ype,xpi,ypi);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               CANCELLA MESH
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

delete('C:\Users\Andrea\Documents\MATLAB\TESI\giustissima.msh');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               TEST FITNESS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fit=[fitness(1) fitness(2)];
fit=fit';

fid=fopen('FIT_TEST.txt','rt');
FIT_TEST=fscanf(fid,'FITNESS %f %f');
fclose(fid);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               PARETO RANKING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CONTROL=(fit<FIT_TEST);
CONTROL=CONTROL';

if CONTROL(1)==1&&CONTROL(2)==1;

    fid=fopen('FIT_TEST.txt','wt');
    fprintf(fid,'FITNESS %f %f',fitness(1),fitness(2));
    fclose(fid);
    fid=fopen('Pareto.txt','at');

fprintf(fid,'*****\r\n'
);

```

```

fprintf(fid, 'NUOVO FRONTE\r\n');

fprintf(fid, '*****\r\n');
fprintf(fid, '-----\r\n');
fprintf(fid, 'FITNESS=%5.5f %5.5f\r\n', fitness(1), fitness(2), ANGOLO);
fprintf(fid, '-----\r\n');
fprintf(fid, 'DNA=%3.5 %3.5 %3.5 %3.5 %3.5 %3.5 %3.5 %3.5 %3.5\r\n', xey);
fprintf(fid, 'XBE=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', xbe);
fprintf(fid, 'YBE=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', ybe);
fprintf(fid, 'XBI=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', xbi);
fprintf(fid, 'YBI=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', ybi);
fprintf(fid, 'CL=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f \r\n', CL);
fprintf(fid, 'CD=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f \r\n', CD);
fclose(fid);

fid=fopen('BackUP.txt', 'at');
fprintf(fid, '-----\r\n');
fprintf(fid, 'FITNESS=%5.5f %5.5f\r\n', fitness(1), fitness(2));
fprintf(fid, '-----\r\n');
fprintf(fid, 'XBE=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', xbe);
fprintf(fid, 'YBE=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', ybe);
fprintf(fid, 'XBI=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', xbi);
fprintf(fid, 'YBI=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', ybi);
fprintf(fid, 'CL=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f \r\n', CL);
fprintf(fid, 'CD=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f \r\n', CD);
fclose(fid);

elseif any(CONTROL)

fid=fopen('Pareto.txt', 'at');
fprintf(fid, '-----\r\n');
fprintf(fid, 'FITNESS=%5.5f %5.5f\r\n', fitness(1), fitness(2), ANGOLO);
fprintf(fid, '-----\r\n');
fprintf(fid, 'XBE=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', xbe);
fprintf(fid, 'YBE=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', ybe);
fprintf(fid, 'XBI=%3.5f %3.5f %3.5f %3.5f %3.5f %3.5f %3.5f\r\n', xbi);

```

```

        fprintf(fid, 'YBI=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', ybi);
        fclose(fid);
        fid=fopen('BackUP.txt', 'at');
        fprintf(fid, '-----
\r\n');
        fprintf(fid, 'FITNESS=%5.5f  %5.5f\r\n', fitness(1), fitness(2));
        fprintf(fid, '-----
\r\n');
        fprintf(fid, 'XBE=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', xbe);
        fprintf(fid, 'YBE=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', ybe);
        fprintf(fid, 'XBI=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', xbi);
        fprintf(fid, 'YBI=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', ybi);
        fclose(fid);
    else
        fid=fopen('BackUP.txt', 'at');
        fprintf(fid, '-----
\r\n');
        fprintf(fid, 'FITNESS=%5.5f  %5.5f\r\n', fitness(1), fitness(2));
        fprintf(fid, '-----
\r\n');
        fprintf(fid, 'XBE=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', xbe);
        fprintf(fid, 'YBE=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', ybe);
        fprintf(fid, 'XBI=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', xbi);
        fprintf(fid, 'YBI=%3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f  %3.5f
\r\n', ybi);
        fclose(fid);
        fid=fopen('Fitnessperpareto.txt', 'at');
        fprintf(fid, '%5.5f  %5.5f\r\n', fitness(1), fitness(2));
        fclose(fid);
    end
end

```

## Appendice C: Routine accessorie

```
function importcd(fileToRead1)
%IMPORTFILE(FILETOREAD1)
% Imports data from the specified file
% FILETOREAD1: file to read

% Auto-generated by MATLAB on 21-May-2013 10:51:43

DELIMITER = ' ';
HEADERLINES = 12;

% Import the file
newData1 = importdata(fileToRead1, DELIMITER, HEADERLINES);

% Create new variables in the base workspace from those fields.
vars = fieldnames(newData1);
for i = 1:length(vars)
    assignin('caller', vars{i}, newData1.(vars{i}));
end
```

# Bibliografia

1. H.K. Versteeg ,An introduction to computational fluid dynamics. Longman scientific & technical.
2. J.H.Holland. Algoritmi genetici. Le Scienze, n.289, 1992
3. L.Davis. Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991
4. J. H. Holland. Adaptation in Natural and Artificial Systems. The University of Michigan Press, 1975
5. M. Mitchell. An Introduction to Genetic Algorithms. MIT Press, 1996
7. Guida Matlab 2010 student version.
8. Theory guide Ansys 14.5 .
9. Appunti dalle lezione di “Moderni metodi di ottimizzazione delle macchine.