

POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria Meccanica



MODELLAZIONE CFD DI MACCHINE VOLUMETRICHE DI TIPO SCROLL

Relatore: Prof. Gianluca MONTENEGRO

Tesi di laurea di:

Stefano Luigi ROSSI – Matricola 820009

Anno Accademico 2014/2015

*“Tutti i nostri sogni possono diventare realtà
se abbiamo il coraggio di perseguirli”*

Walt Disney

RINGRAZIAMENTI

Sono seduto alla mia scrivania intento a scrivere questa tesi e mi viene in mente una frase: “Dottore Magistrale in Ingegneria Meccanica”... detta così fa quasi paura ma è il titolo che mi spetta dopo questi anni di duro lavoro, e non potrei essere più felice!

Forse non me ne sono ancora reso conto, ma questo secondo traguardo è quello veramente importante, quello che chiude la mia carriera universitaria e con essa la mia vita da studente.

Ringrazio prima di tutti il Professore Montenegro per avermi guidato ed aiutato nello sviluppo di questo lavoro di tesi, e soprattutto per avermi sopportato durante questi mesi.

Ringrazio gli amici che mi hanno accompagnato in questo percorso, per la loro compagnia durante le lunghissime ore di lezione e di studio.

Ringrazio anche tutti quelli che, inconsapevolmente, con i loro “eh... Ingegneria è dura!” mi hanno spinto a non farmi sopraffare dalle difficoltà.

E l'ultimo grazie, il più grande di tutti, è per i miei genitori, senza i quali non sarei mai potuto arrivare fino a questo punto. E non parlo solo del sostegno economico, ma di tutti quegli aiuti taciti ed espliciti che vengono dal cuore e che mi hanno permesso di superare le ansie, le difficoltà ed i momenti di sconforto.

Grazie infinite... ancora... e per sempre!

INDICE

Sommario	IX
Abstract	XI
Introduzione: Scopo e struttura della tesi	1
Capitolo 1: Macchine volumetriche scroll	5
1.1 Funzionamento come compressore	8
1.2 Funzionamento come espansore	10
1.3 Costruzione e caratteristiche	11
1.4 Vantaggi e svantaggi	13
Capitolo 2: <i>OpenFOAM</i>	15
2.1 Pre – processing	16
2.2 Solving	17
2.3 Post – processing	19
2.4 <i>LibRotaryMachine</i>	19
Capitolo 3: Fluidodinamica computazionale	23
3.1 Equazioni fondamentali	24
3.1.1 Conservazione della massa	27
3.1.2 Conservazione della quantità di moto	28
3.1.3 Conservazione dell'energia	30
3.1.4 Equazione di stato	31
3.2 Metodo dei volumi finiti	32
3.2.1 Discretizzazione del dominio	33
3.2.2 Discretizzazione delle equazioni	37
Capitolo 4: Caso 3D	43
4.1 Creazione della geometria: 1° metodo	43
4.2 Creazione della geometria: 2° metodo	52
4.3 Simulazione del funzionamento	55
4.3.1 <i>rotaryMachineDynamicSetUp</i>	55
4.3.2 <i>runMultiMeshCase</i>	58

Capitolo 5: Analisi di sensitività ai <i>gap</i>	59
Capitolo 6: Risultati delle prove	63
6.1 Condizioni al contorno	63
6.2 Caratteristiche del fluido	64
6.3 Risultati	66
Capitolo 7: Caso 2D	73
7.1 Calcolo della densità	75
7.2 Calcolo dell'entalpia	76
7.3 Verifica di funzionamento	81
Sviluppi futuri	87
Allegati	89
Allegato 1: Dizionario <i>blockMeshDict</i>	91
Allegato 2: Dizionario <i>snappyHexMeshDict</i>	95
Allegato 3: Dizionario <i>extrudeMeshDict</i>	103
Allegato 4: Script <i>CreateBaseMesh</i>	105
Bibliografia	107

INDICE DELLE FIGURE

1.1	Profilo della spirale	5
1.2	Spirali di una macchina scroll	7
1.3	Fasi del processo di compressione	8
1.4	Fasi del processo di espansione	10
1.5	Vista in sezione di un compressore scroll	12
2.1	Struttura di <i>OpenFOAM</i>	15
2.2	Struttura della libreria <i>LibRotaryMachine</i>	20
2.3	Struttura di un caso	21
3.1	Esempio di <i>Mesh Structured</i>	35
3.2	Esempio di <i>Mesh Unstructured</i>	35
3.3	Dettaglio della faccia f in comune a due celle adiacenti	39
3.4	Interpolazione del valore ϕ_f sulla faccia f	39
4.1	Vista in sezione del rotore	43
4.2	Vista 3D del rotore	44
4.3	Vista 3D dello statore	44
4.4	Vista 3D del gruppo statore e rotore	46
4.5	Vista 3D della <i>Mesh</i> di calcolo con 415.000 celle	47
4.6	Trafilamento attraverso il <i>gap</i> assiale (a sinistra) e radiale (a destra)	48
4.7	Esempio di macchina sigillata	49
4.8	Vista 3D della <i>Mesh</i> di calcolo semplificata con 6.800 celle	50
4.9	Vista 3D della <i>Mesh</i> finale	51
4.10	Vista 3D del condotto di scarico realizzato con l'utility <i>BlockMesh</i>	52
4.11	Vista 3D dell'insieme <i>duct</i> e superficie <i>top</i>	53
4.12	Risultato del processo di <i>Snap</i>	53
4.13	Confronto tra la qualità delle celle in corrispondenza degli spigoli ottenute con il primo metodo (a sinistra) e con il secondo metodo (a destra)	54
4.14	Esempio di cella con <i>skewness</i> nullo (a sinistra) e con <i>skewness</i> elevato (a destra)	56
4.15	Schema di funzionamento dell'utility <i>rotaryMachineDynamicSetUp</i>	57
5.1	Esempio di <i>refinement</i> della <i>Mesh</i> tra due superfici vicine	59
6.1	Campo di pressione all'inizio della simulazione (1°)	66
6.2	Campo di densità all'inizio della simulazione (1°)	66
6.3	Campo di temperatura all'inizio della simulazione (1°)	67
6.4	Campo di velocità all'inizio della simulazione (1°)	67
6.5	Campo di pressione a 50° di rotazione	68

6.6	Campo di densità a 50° di rotazione	68
6.7	Campo di temperatura a 50° di rotazione	69
6.8	Campo di velocità a 50° di rotazione	69
6.9	Particolare della velocità nella zona di trafilamento	70
7.1	Vista del profilo di statore e rotore usato per caso 2D	74
7.2	Vista della <i>Mesh</i> semplificata usata nel caso 2D	74
7.3	Campo di pressione all'inizio del movimento	82
7.4	Campo di densità all'inizio del movimento	82
7.5	Campo di temperatura all'inizio del movimento	83
7.6	Campo di velocità all'inizio del movimento	83
7.7	Campo di pressione a 90° di rotazione	84
7.8	Campo di densità a 90° di rotazione	84
7.9	Campo di temperatura a 90° di rotazione	85
7.10	Campo di velocità a 90° di rotazione	85

INDICE DELLE TABELLE

5.1	Numero di celle e tempo di generazione della <i>Mesh</i> al variare del <i>gap</i> radiale	60
5.2	Numero di celle e tempo di generazione della <i>Mesh</i> al variare del <i>gap</i> assiale	61
6.1	Coefficienti NIST per il calcolo del calore specifico dell'aria	65
7.1	Esempio di coefficienti per il calcolo dell'energia di <i>Helmholtz</i> ideale	78
7.2	Esempio di coefficienti per il calcolo dell'energia di <i>Helmholtz</i> ideale	79
7.3	Esempio di coefficienti per il calcolo dell'energia di <i>Helmholtz</i> reale	80

INDICE DEI GRAFICI

5.1	Influenza del <i>gap</i> radiale sul numero di celle nella <i>Mesh</i>	60
5.2	Influenza del <i>gap</i> assiale sul numero di celle nella <i>Mesh</i>	61
5.3	Tempo di generazione della <i>Mesh</i> al variare del numero di celle	62
6.1	Andamento della pressione allo scarico	71
6.2	Andamento della temperatura allo scarico	72

SOMMARIO

L'obiettivo di questa tesi è quello di effettuare una simulazione CFD del funzionamento di una macchina volumetrica di tipo Scroll utilizzando il software *OpenFOAM*.

Il lavoro ha riguardato la creazione delle geometrie 3D che compongono la macchina e la riproduzione del movimento del rotore considerando l'effetto del gas reale.

La geometria reale della macchina è stata realizzata utilizzando un software CAD e riferendosi a parametri geometrici reperibili in letteratura.

Il movimento della griglia di calcolo viene gestito attraverso delle applicazioni create appositamente, che permettono di riprodurre la rotazione della macchina creando una successione di *Mesh* tenendo conto della loro qualità.

Come fluido di lavoro è stata scelta aria e, per considerarne il comportamento reale, le sue proprietà termodinamiche sono calcolate utilizzando i valori forniti dal database NIST.

Alla fine, considerando una geometria semplificata, il risolutore è stato impostato per poter utilizzare specifici modelli termodinamici per il calcolo della densità e dell'entalpia del fluido.

Parole chiave: macchina Scroll, *OpenFoam*, CFD, fluido reale.

ABSTRACT

The goal of this thesis is to make a CFD simulation of the operation of a volumetric Scroll machine using the software *OpenFOAM*.

The work involved the creation of 3D geometry that make up the machine and the reproduction of the movement of the rotor considering the effect of real gas.

The real geometry of the machine was designed using a CAD software and referring to geometrical parameters found in literature.

The movement of the computing grid is managed through specifically created applications that allow to reproduce the rotation of the machine creating a succession of *Meshs* taking into account their quality.

The working fluid chosen was air and, to consider the real behavior, its thermodynamic properties were calculated using the values provided by the NIST database.

At the end, by considering a simplified geometry, the solver was set to use specific thermodynamic models for the calculation of the density and enthalpy of the fluid.

Key words: Scroll machine, *OpenFoam*, CFD, real fluid.

INTRODUZIONE

SCOPO E STRUTTURA DELLA TESI

Negli ultimi anni la fluidodinamica computazionale, meglio conosciuta con l'acronimo CFD, ha subito una rapidissima espansione e oggi viene sempre più usata nell'analisi del moto dei fluidi, in particolare all'interno di macchine volumetriche quali compressori, turbine e motori a combustione interna.

Una volta modellato il problema da analizzare e definiti i modelli matematici più opportuni, grazie all'uso della fluidodinamica computazionale è possibile simulare il comportamento reale di un sistema fluido senza ricorrere a prove sperimentali in laboratorio, riducendo così i tempi necessari all'ottenimento dei risultati.

In ambito industriale i tempi di calcolo sono molto importanti; l'analisi CFD viene quindi effettuata introducendo delle semplificazioni che riducono la complessità del fenomeno reale al fine di diminuire gli oneri computazionali senza limitare la precisione della soluzione ottenuta.

In questo lavoro di tesi la CFD viene impiegata per simulare il funzionamento di un compressore di tipo scroll, ovvero una macchina volumetrica che non è caratterizzata da una geometria puramente assiale o radiale come i classici compressori, ma è l'azione combinata di due spirali ad effettuare la compressione del fluido.

Le macchine scroll vengono impiegate soprattutto negli impianti di refrigerazione e condizionamento sia civili che industriali e sono particolarmente apprezzate per la loro silenziosità e uniformità di funzionamento.

Per eseguire la simulazione del compressore la geometria tridimensionale reale della macchina viene riprodotta con l'utilizzo di un software CAD e, successivamente, usando il software CFD *OpenFOAM*, viene generato il movimento della macchina e definite le caratteristiche fisiche e termodinamiche del fluido di lavoro.

La struttura *open source* di *OpenFOAM* è molto utile perché rende possibile la modifica di tutte le parti del programma per adattarlo allo studio del problema in esame.

Il lavoro di tesi è stato così strutturato:

Capitolo 1

Descrizione delle caratteristiche geometriche delle macchine volumetriche scroll e analisi del loro funzionamento quando sono configurate sia in compressione che in espansione.

Capitolo 2

Descrizione della struttura e del funzionamento del software *OpenFOAM*, con le sue utility ed i risolutori che permettono di impostare ed eseguire la simulazione della macchina.

Capitolo 3

Dedicato alla fluidodinamica computazionale ed all'analisi di: relazioni matematiche che sono alla base di un problema CFD, metodi di creazione e di discretizzazione spaziale e temporale della griglia di calcolo (*Mesh*), tecniche di discretizzazione delle equazioni.

Capitolo 4

Analizza la geometria reale della macchina, spiegando come è stata ottenuta e come vengono utilizzate le utility di *OpenFOAM* per realizzare il movimento della girante.

Capitolo 5

Analisi di sensitività ai *gap* presenti nella macchina, ovvero descrizione di come la distanza assiale e radiale tra statore e rotore influenza il numero di celle presenti nella *Mesh* ed i tempi computazionali richiesti per la sua generazione.

Capitolo 6

Descrizione delle proprietà fisiche e termodinamiche del fluido di lavoro utilizzato, definizione delle condizioni al contorno e analisi dei risultati della simulazione.

Capitolo 7

Considera una geometria solamente bidimensionale e descrive come il risolutore di *OpenFOAM* viene modificato per renderlo adatto ad utilizzare particolari modelli termodinamici per il calcolo della densità e dell'entalpia del fluido.

CAPITOLO 1

MACCHINE VOLUMETRICHE SCROLL

Il concetto di macchina volumetrica di tipo *Scroll*, detta anche *a spirale orbitale*, è stato pensato e brevettato per la prima volta dall'ingegnere francese Léon Creux nel 1905 ma, a causa delle tecnologie dell'epoca che non permettevano di ottenere le limitate tolleranze necessarie al loro corretto funzionamento, queste macchine hanno iniziato a diffondersi solo dopo la seconda guerra mondiale e sono state rese disponibili sul mercato per applicazioni industriali e residenziali a partire dalla fine degli anni ottanta.

A seconda del loro senso di rotazione, le macchine scroll permettono di effettuare la compressione o l'espansione di un gas grazie all'azione combinata di due spirali realizzate con un profilo detto *ad evolvente di cerchio*, ovvero ogni punto della spirale ha una distanza r dal centro pari al prodotto tra una costante arbitraria a e l'angolo ϑ

$$r = a \vartheta \quad (1.1)$$

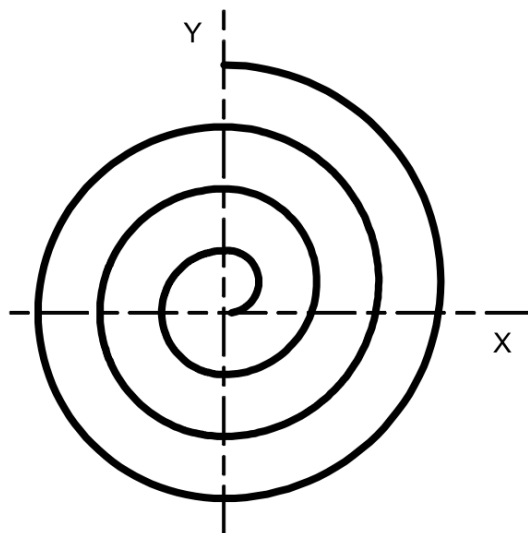


Figura 1.1 – Profilo della spirale

Considerando una geometria tridimensionale, le spirali di una macchina scroll sono costituite essenzialmente da una striscia di metallo avvolta su se stessa, che viene chiusa ad una estremità dalla superficie piana a cui essa è vincolata.

Una delle due spirali di cui è composta una macchina scroll, generalmente quella superiore, è rigidamente collegata alla struttura esterna della macchina in modo da rimanere fissa e quindi costituire lo statore.

La spirale inferiore invece, che rappresenta il rotore della macchina, è collegata al motore elettrico tramite un albero eccentrico che ne garantisce il movimento lungo una traiettoria orbitale attorno allo statore.

Le due spirali sono realizzate con la medesima geometria e questo rende molto semplice il loro accoppiamento che viene effettuato posizionandole una dentro l'altra con un angolo relativo di 180° in modo tale che, durante il movimento, i due profili si avvicinino tra loro in più punti creando delle sacche di gas che vengono progressivamente compresse o espanse a seconda della direzione del movimento orbitale.

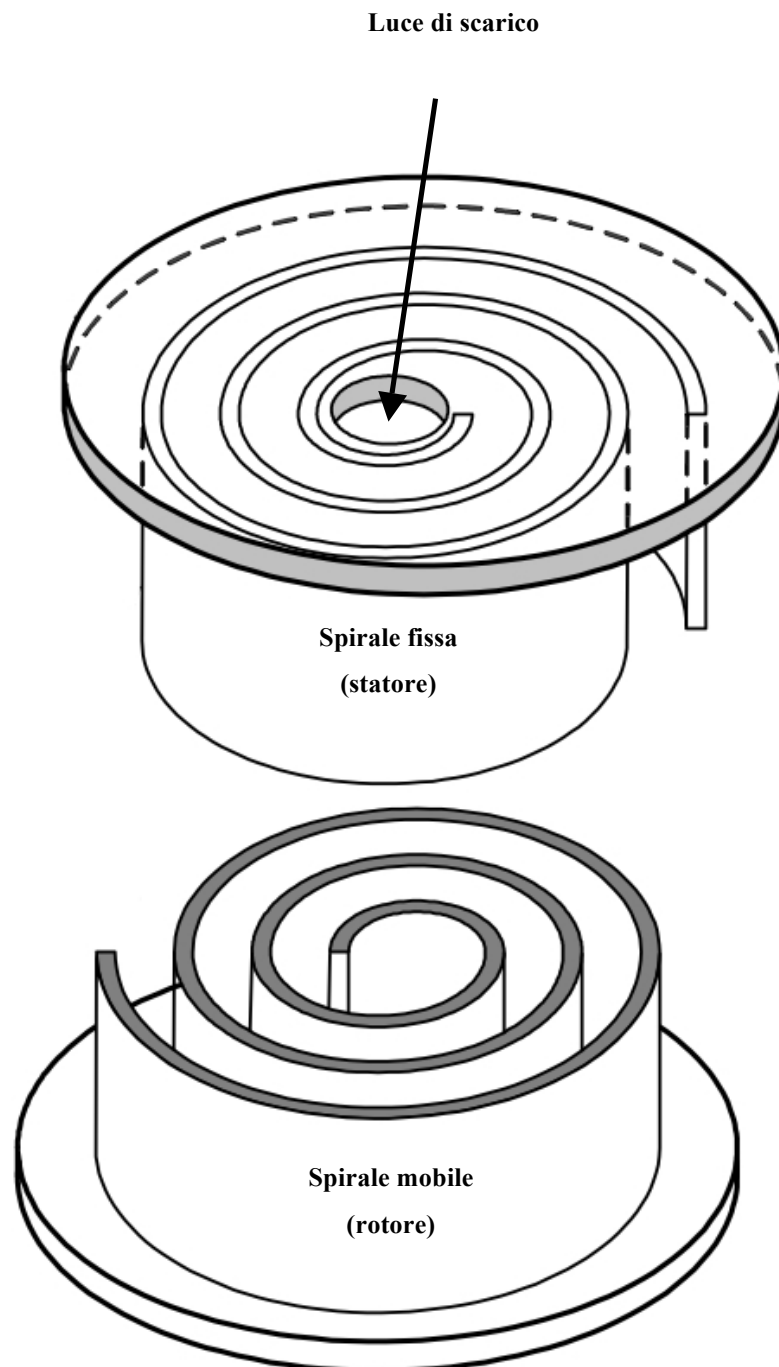


Figura 1.2 – Spirali di una macchina scroll

1.1 – FUNZIONAMENTO COME COMPRESSORE

Generalmente, quando una macchina scroll è configurata per funzionare come un compressore, le spirali sono disegnate in modo da estendersi su un angolo compreso tra i 1000° ed i 1200° ed il processo di compressione del gas si completa in tre movimenti orbitali.

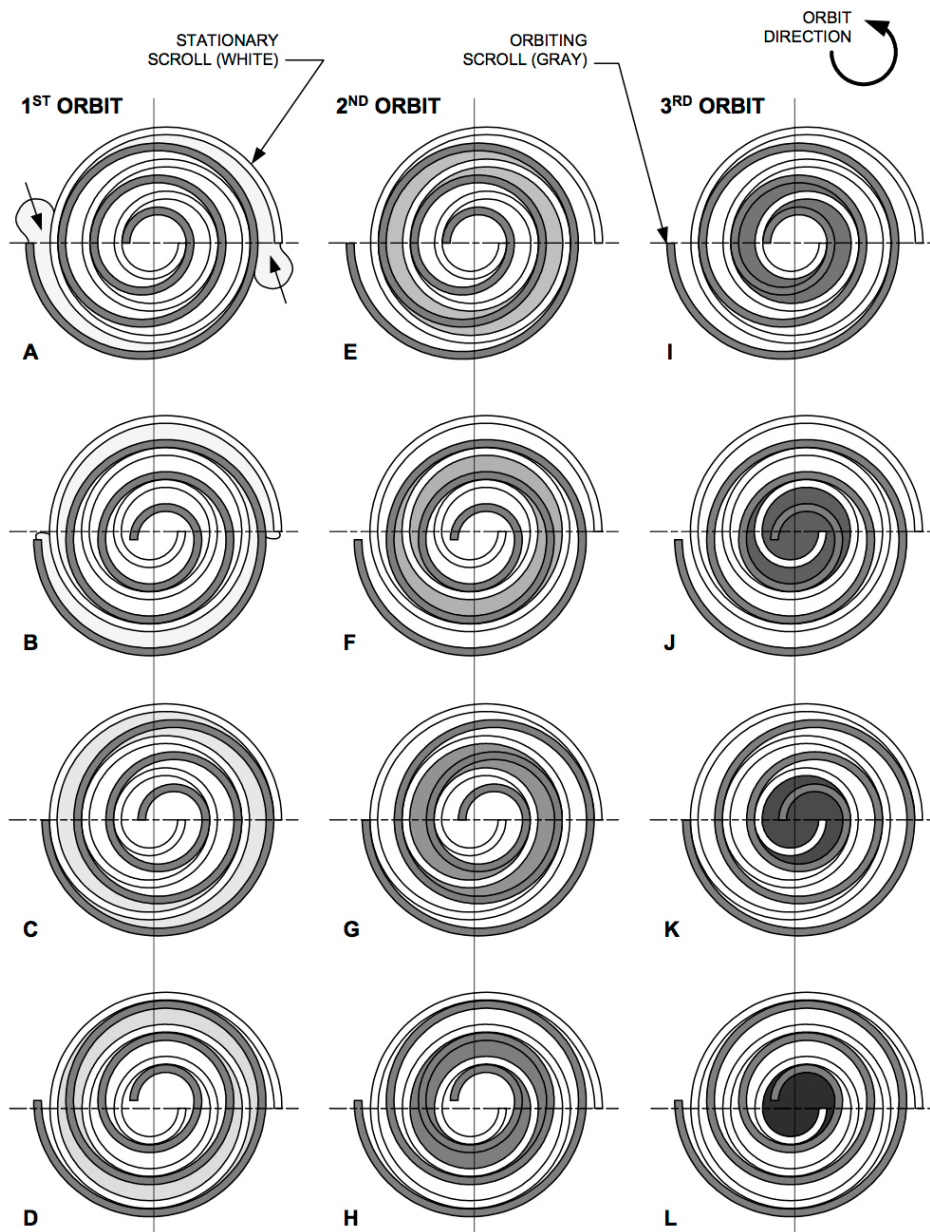


Figura 1.3 – Fasi del processo di compressione

La prima orbita (figure 1.3 A, 1.3 B, 1.3 C, 1.3 D) inizia con le estremità delle spirali completamente aperte; in questa posizione l'interno del compressore viene messo in comunicazione con l'ambiente esterno ed il gas viene aspirato andando a riempire lo spazio compreso tra i profili delle due spirali.

Proseguendo con il movimento eccentrico della spirale mobile rispetto a quella fissa, le aperture alle estremità delle spirali vengono progressivamente chiuse con conseguente interruzione dell'aspirazione dall'ambiente esterno e generazione di una sacca di gas che riempie una parte della spirale.

Dal momento che il profilo delle due spirali è *a evolvente*, durante il movimento il punto di tangenza tra i due profili si sposta dalla periferia al centro della spirale, consentendo al secondo movimento orbitale (figure 1.3 E, 1.3 F, 1.3 G, 1.3 H) di spingere la sacca di gas sempre più verso l'interno imponendole una costante riduzione di volume e quindi una compressione.

Contemporaneamente alla seconda orbita, le estremità delle spirali si aprono nuovamente dando inizio ad una ulteriore fase di aspirazione.

Il movimento prosegue con la terza orbita (figure 1.3 I, 1.3 J, 1.3 K, 1.3 L) che comprime sempre di più il gas e lo guida verso il centro della spirale.

Alla fine dei tre movimenti orbitali la sacca di gas, contenuta tra i profili delle due spirali, raggiunge il centro della macchina ovvero la zona con volume minimo; il fluido al suo interno giunge così al valore massimo di pressione e viene espulso attraverso la luce di scarico al centro dello statore.

Le caratteristiche geometriche delle spirali, unitamente al movimento orbitale ed eccentrico del rotore, permettono di avere, durante il funzionamento, la compressione simultanea di più sacche di gas distribuite lungo il profilo della spirale, consentendo di realizzare un processo uniforme che permette di limitare le pulsazioni, le vibrazioni ed il rumore prodotto.

1.2 – FUNZIONAMENTO COME ESPANSORE

Una macchina scroll può essere anche configurata per funzionare come un espansore. Questa configurazione viene facilmente realizzata partendo da un compressore ed invertendone il senso di rotazione, in modo che anche il moto del fluido all'interno della macchina venga invertito.

I vantaggi rimangono gli stessi della configurazione precedente.

Questo tipo di espansore è utilizzabile solo in impianti che producono una potenza limitata, tipicamente inferiore a 1 kW.



Figura 1.4 – Fasi del processo di espansione

1.3 – COSTRUZIONE E CARATTERISTICHE

Per quanto riguarda la loro costruzione, nella maggiore parte dei casi, tutte le parti che costituiscono un compressore scroll vengono disposte in verticale all'interno di un involucro ermetico di metallo.

Alla base della struttura trovano posto il motore elettrico, la pompa dell'olio, le parti che permettono il movimento del rotore ed il condotto di aspirazione del gas. Il fluido a bassa pressione aspirato dall'ambiente viene usato anche per il raffreddamento delle componenti della macchina.

La parte superiore dell'involucro contiene le due spirali e la luce di scarico del gas ad alta pressione.

Il motore elettrico che aziona la macchina è collegato ad un inverter che consente di regolare la velocità di rotazione e, di conseguenza, la potenza del compressore.

Per evitare una eccessiva usura durante il movimento, le spirali devono essere realizzate rispettando ristrettissime tolleranze geometriche e devono avere una rugosità superficiale molto limitata, in genere inferiore a $0,7 \mu m$.

Le macchine scroll hanno una efficienza molto elevata perché ognuna di esse viene dimensionata e ottimizzata in funzione delle caratteristiche del fluido di lavoro che sarà utilizzato; in questo modo si riescono ad avere prestazioni elevate in tutte le condizioni di funzionamento sia a pieno carico che a carichi parziali.

La particolare geometria di queste macchine ha anche il vantaggio di rendere superfluo l'utilizzo della valvola di aspirazione e, a seconda dell'applicazione, si può ricorrere o meno all'uso di una valvola di scarico dinamica che, da un lato, è benefica perché permettere al compressore di coprire un ampio campo di condizioni operative ma, al tempo stesso, è dannosa perché introduce delle perdite di carico che riducono l'efficienza della macchina.

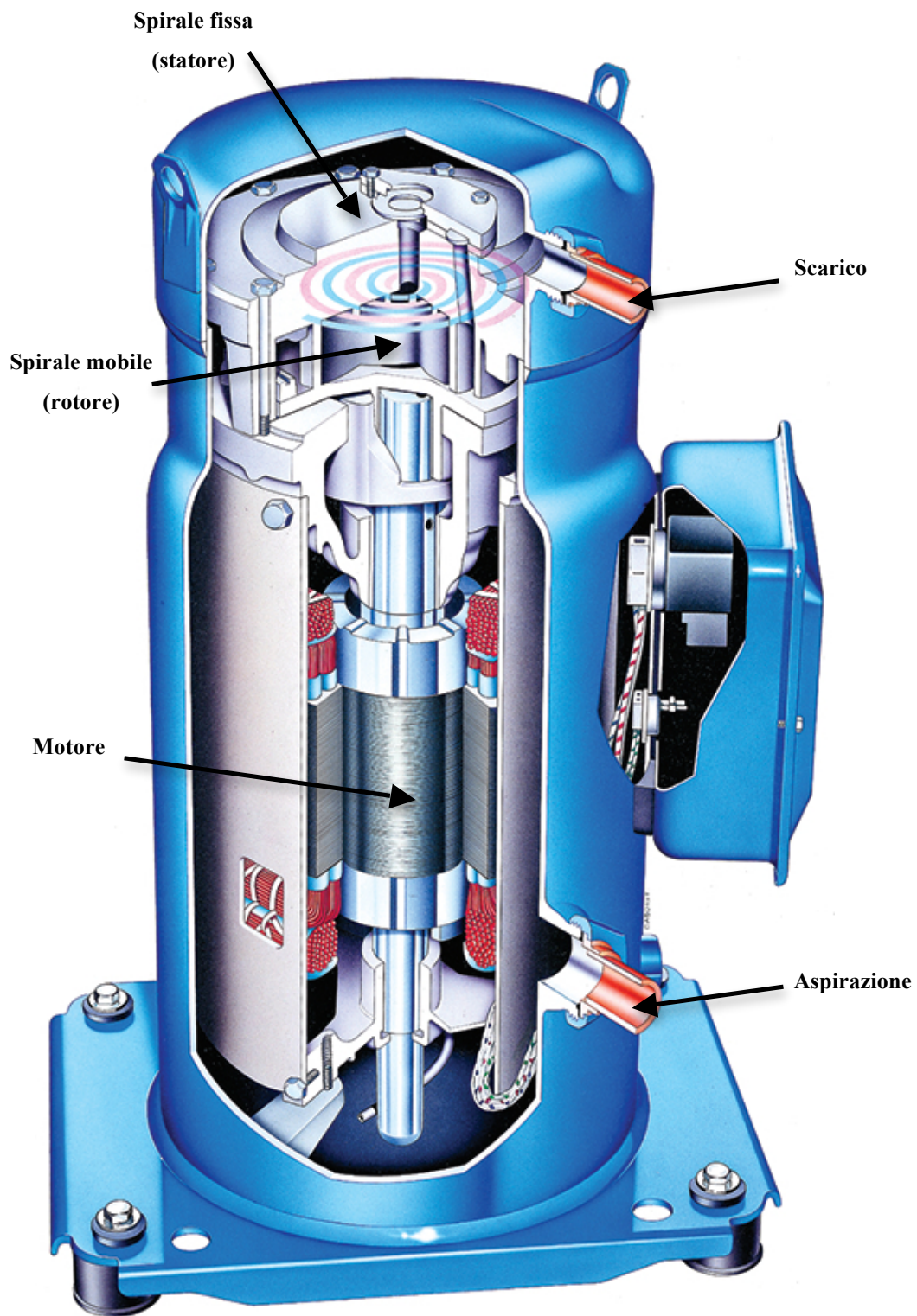


Figura 1.5 – Vista in sezione di un compressore scroll

A seconda delle loro dimensioni e del numero di unità abbinate, questi compressori permettono di ottenere potenze massime che vanno dai 15 – 20 kW per applicazioni residenziali, fino ad arrivare a 1000 – 1500 kW e oltre per applicazioni industriali. Le macchine scroll vengono utilizzate principalmente negli impianti di refrigerazione, di condizionamento, nelle pompe di calore e nei sistemi di produzione di aria compressa.

1.4 – VANTAGGI E SVANTAGGI

Se confrontati con i compressori alternativi tradizionali, le macchine scroll presentano una serie di vantaggi:

- Permettono di realizzare un processo di compressione molto uniforme che consente di ottenere un ottimo bilanciamento dinamico, riduzione delle vibrazioni, riduzione delle pulsazioni di pressione in aspirazione e allo scarico, attenuazione del rumore prodotto.
- L'assenza delle valvole di aspirazione e di scarico contribuisce ad avere una elevata efficienza sia a pieno carico che a carichi parziali.
- Pesi e ingombri sono inferiori rispetto a quelli dei compressori alternativi di pari potenza.
- Hanno poche parti in movimento con conseguente riduzione della probabilità di guasto.
- Le ristrette tolleranze con cui vengono realizzati permettono di avere un limitato attrito tra le parti in movimento riducendo l'usura.
- L'assenza dello *spazio nocivo* (volume della camera di compressione in cui rimane una parte di gas compresso che non viene espulso in fase di mandata) permette di ottenere un rendimento volumetrico prossimo al 100%.

Unitamente a questi aspetti positivi, i compressori scroll hanno anche delle caratteristiche negative:

- In direzione assiale e radiale le due spirali non sono a contatto, ma tra di loro sono presenti dei giochi attraverso i quali si ha il trafileamento del fluido con conseguente riduzione delle prestazioni.
- Sono molto vulnerabili alla presenza di detriti nel fluido, perché durante il funzionamento questi vengono premuti tra le due spirali danneggiandole.
- Le elevate forze centrifughe che si sviluppano durante il funzionamento limitano le dimensioni massime della spirale e quindi il valore di pressione che è possibile ottenere.
- Sono macchine realizzate a tenuta stagna e per questo non possono essere facilmente smontate sul campo per effettuare operazioni di manutenzione.

CAPITOLO 2

OpenFOAM

Per eseguire le simulazioni di funzionamento del compressore scroll è stato utilizzato il software *OpenFOAM*, acronimo di *Open source Field Operation And Manipulation*, nella release 2.3.0.

OpenFOAM è un software sviluppato per ambienti Linux composto da una serie di classi scritte in linguaggio C++ che, una volta compilate, generano delle librerie che vengono utilizzate dalle varie applicazioni di cui è composto il programma per simulare il problema in esame.

Il programma è stato concepito per la prima volta nel 1993 facendo riferimento alla meccanica dei continui e per questo motivo può essere impiegato in un grande numero di applicazioni, ma soprattutto per eseguire simulazioni fluidodinamiche.

Il software, come dice il nome, è fornito con licenza *open source*, ovvero può essere liberamente scaricato e modificato in tutte le sue parti.

Avere una licenza di questo tipo significa che chiunque può agire sul sorgente del software modificandolo, migliorandolo e rendendolo adatto alle proprie esigenze senza dipendere da fornitori esterni ed in modo del tutto gratuito, cosa che non sarebbe possibile con i normali software commerciali.

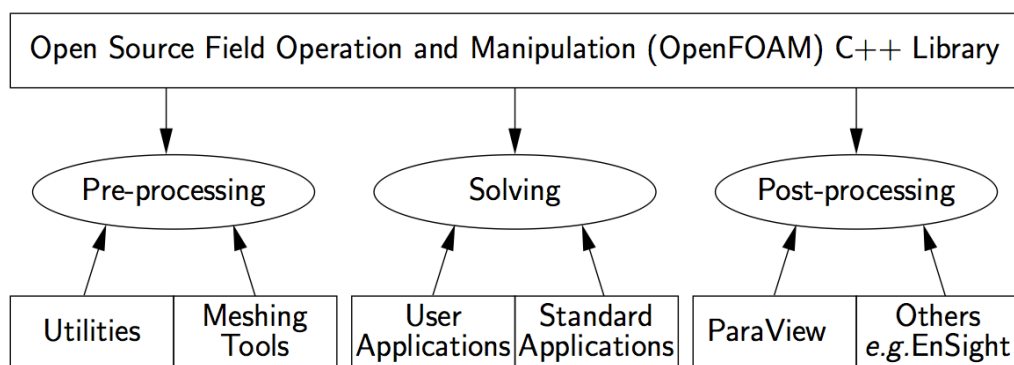


Figura 2.1 – Struttura di *OpenFOAM*

Come tutti i codici CFD, anche *OpenFOAM* è composto da 3 ambienti principali:

Ambiente di pre-processing

Corrisponde alla fase iniziale in cui viene preparato il problema per la simulazione.

Risolutore

In cui viene eseguita la simulazione.

Ambiente di post-processing

Permette, una volta eseguita la simulazione, di analizzare i risultati ottenuti.

2.1 – PRE PROCESSING

L'ambiente di pre-processing è un insieme di applicativi, detti *utilities*, che permettono di definire tutti i parametri legati al problema oggetto della simulazione.

In particolare, vengono richieste:

- La definizione del dominio di calcolo, ovvero la regione geometrica di interesse al problema in cui verranno risolte le equazioni.
- La divisione del dominio in sotto-domini più piccoli, ovvero la generazione della griglia di calcolo, detta *Mesh*, più opportuna alla risoluzione del problema.
- La definizione, attraverso opportuni modelli, dei fenomeni fisici e chimici che intervengono nel problema.
- La definizione delle proprietà fisiche e termodinamiche del fluido di lavoro.
- L'applicazione di ipotesi semplificative del fenomeno reale in modo da rendere più semplice e agevole la simulazione CFD.

- L'implementazione di opportune condizioni al contorno che identificano i confini del dominio di calcolo e le caratteristiche che il fenomeno studiato assume in quelle zone.

Se il problema analizzato dipende anche dal tempo, le condizioni al contorno si aggiungono anche delle condizioni iniziali che descrivono lo stato del fenomeno all'istante iniziale della simulazione.

2.2 – SOLVING

Per effettuare una simulazione CFD si utilizzano delle applicazioni, dette *solutori*, che sono degli eseguibili già presenti all'interno del programma e che, all'occorrenza, possono essere modificati o ricreati da zero dall'utente.

Per risolvere i problemi legati alla fluidodinamica esistono principalmente tre tecniche differenti che permettono di semplificarne la risoluzione: le equazioni alle derivate parziali vengono convertite in problemi algebrici.

Il metodo delle differenze finite

Permette di approssimare una generica funzione ϕ incognita usando il valore che essa assume in determinati punti tramite il suo sviluppo in serie di Taylor. Questo metodo è stato il primo ad essere utilizzato nella CFD però è poco flessibile e quindi non adatto a risolvere problemi con domini irregolari o con condizioni al contorno complesse; oggi viene usato soltanto in alcuni codici specializzati.

Il metodo degli elementi finiti

Questo metodo, noto brevemente con l'acronimo FEM, utilizza delle semplici funzioni (lineari o quadratiche) per descrivere la variazione locale della funzione ϕ incognita in tutti i punti della griglia di calcolo. Contrariamente al metodo precedente, è molto adatto alla risoluzione di problemi con un dominio molto complesso o variabile nel tempo, e per questo viene impiegato principalmente nell'analisi strutturale e nei problemi fluidodinamici con numeri di Reynolds elevati.

Il metodo dei volumi finiti

È l'approccio più usato nei codici CFD, *OpenFOAM* compreso, e ha una formulazione intermedia tra i primi due metodi, con lo svantaggio però di non essere adatto a simulare sistemi discontinui. Con questo metodo le equazioni che governano il moto del fluido vengono integrate, discretizzate e risolte in tutti i volumi di controllo in cui è stato diviso il dominio di calcolo.

La precisione di una simulazione CFD dipende direttamente dal numero di celle di cui è composta la griglia di calcolo; avere una *Mesh* più fitta significa quindi riuscire ad ottenere una soluzione più accurata, a scapito però dei tempi di calcolo.

Per riuscire ad avere risultati soddisfacenti senza avere un aumento eccessivo dei tempi computazionali si usano spesso griglie di calcolo non uniformi, più fitte nelle zone dove si ha la variazione maggiore delle grandezze di interesse e più lasche nelle restanti zone del dominio.

Le tecniche di discretizzazione utilizzate per l'approssimazione delle equazioni, i modelli impiegati per descrivere i fenomeni fisici e le ipotesi semplificative formulate nella fase di pre-processing sono delle approssimazioni del fenomeno reale, e quindi l'analisi CFD restituirà dei risultati che saranno più o meno precisi a seconda della bontà dei modelli utilizzati.

Affinché i risultati forniti dall'analisi CFD possano essere considerati attendibili, i modelli e gli algoritmi utilizzati devono rispettare tre proprietà fondamentali.

Convergenza

Il metodo numerico utilizzato deve essere in grado di avvicinarsi alla soluzione esatta quando le dimensioni delle celle della griglia di calcolo diventano infinitesime.

Consistenza

Le equazioni algebriche ottenute con le tecniche di discretizzazione devono essere equivalenti alle equazioni differenziali originarie quando le dimensioni delle celle della griglia di calcolo diventano infinitesime.

Stabilità

Il metodo numerico non deve accrescere gli errori durante l'esecuzione della simulazione.

2.3 – *POST PROCESSING*

Il post-processing è anch'esso un ambiente formato da un insieme di *utilities* che, una volta risolto il problema fluidodinamico, permette di analizzare i risultati ottenuti.

In particolare è possibile:

- Visualizzare il dominio di calcolo e la relativa *Mesh*.
- Generare grafici 2D e 3D che riassumono i risultati delle simulazioni.
- Simulare l'evoluzione del fenomeno nel tempo all'interno del dominio di calcolo.

Unitamente ad *OpenFOAM* viene fornita l'utility *ParaView* che si occupa quasi interamente di tutta la fase di post-processing.

2.4 – *LibRotaryMachine*

Per la simulazione del compressore scroll, ed in generale delle altre tipologie di macchine rotative, in parallelo al software *OpenFOAM* è stata utilizzata la libreria *LibRotaryMachine* nella versione 2.3.0 sviluppata dal Dipartimento di Energia del Politecnico di Milano.

Questa libreria contiene una serie di classi, *utilities* ed applicazioni necessarie alla simulazione della geometria e del movimento della macchina, ed è dotata anche di modelli che permettono di considerare le caratteristiche ed il comportamento del gas reale.

La libreria è composta da quattro directory principali, ognuna delle quali è a sua volta divisa in sotto-directory con funzioni specifiche

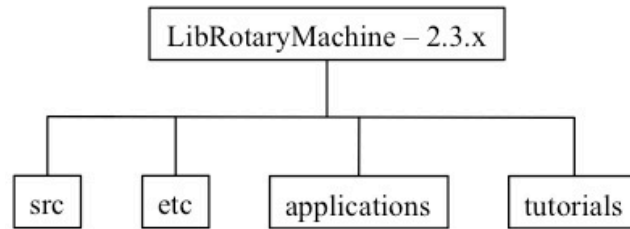


Figura 2.2 – Struttura della libreria *LibRotaryMachine*

Directory *src*

È l'insieme di tutte le classi necessarie allo sviluppo delle utilities e dei solutori.

In particolare questa directory è composta da una parte, denominata *rotaryMachine*, che contiene le classi necessarie a ridefinire il tempo passando da secondi ad angoli ed alla gestione del movimento della *Mesh*, ed un'altra parte, denominata *thermophysicalModels*, che contiene i modelli termodinamici necessari alla descrizione del comportamento del gas reale.

Directory *etc*

Contenete informazioni necessarie alla compilazione ed i dati termodinamici del fluido.

Directory *applications*

Racchiude tutte le utilities ed i solutori necessari ad eseguire tutte le fasi del processo di simulazione: pre-processing, solving e post-processing.

Directory *tutorials*

Contenente i tutorials che permettono di risolvere il problema con differenti strategie.

Ogni problema, definito sinteticamente *caso*, può essere schematizzato nel modo seguente:

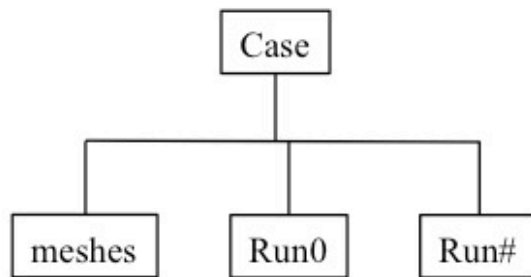


Figura 2.3 – Struttura di un caso

Nella cartella *meshes* sono contenuti i file in formato STL che rappresentano la geometria di statore e rotore, i parametri geometrici necessari alla creazione della griglia di calcolo e le informazioni per la generazione della *Mesh*.

Viene poi creato un numero arbitrario di cartelle *Run* in modo da poter simulare differenti condizioni operative della stessa macchina.

CAPITOLO 3

FLUIDODINAMICA COMPUTAZIONALE

La fluidodinamica computazionale, indicata brevemente con l'acronimo inglese CFD (*Computational Fluid Dynamics*), è una pratica che permette lo studio, tramite calcolatore dei problemi della fluidodinamica, utilizzando tecniche numeriche per la risoluzione delle equazioni matematiche alla base di questi processi.

La diffusione della CFD ha avuto inizio intorno agli anni '60, quando questa tecnica risolutiva fece la sua prima comparsa nei reparti di progettazione, di ricerca e sviluppo in aziende aeronautiche e motoristiche.

A partire dai primi anni '90, grazie alla disponibilità di calcolatori dalle prestazioni sempre più elevate e di software facilmente utilizzabili, l'uso della CFD ha avuto una rapida espansione ed oggi viene impiegata in diversi settori, ad esempio:

- Analisi delle prestazioni di profili alari, in termini di portanza e di resistenza, sia in campo aeronautico che automobilistico.
- Studio dell'idrodinamica delle imbarcazioni e dell'interazione dell'acqua con le strutture off-shore.
- In ambito biomedico per simulare il flusso sanguigno nelle vene e nelle arterie.
- Studio del rumore e propagazione delle onde sonore.
- Simulazione dei processi di raffreddamento in apparati elettrici.
- Analisi dei flussi nei motori a combustione interna e attraverso i canali statorici e rotorici delle turbomacchine.

Grazie all'utilizzo della CFD si riduce sempre più la necessità di ricorrere a prove sperimentali che, in alcuni casi, possono anche essere eliminate, permettendo così di avere:

- una diminuzione dei costi legati alla ricerca ed allo sviluppo
- la possibilità di generare grandi quantità di risultati per le più svariate condizioni operative senza avere nessun aggravio economico
- l'opportunità di eseguire simulazioni in condizioni rischiose che non possono essere realizzate con le normali prove in laboratorio.

Oltre a questo, grazie ai moderni calcolatori, il tempo richiesto da un'analisi CFD risulta essere molto inferiore a quello necessario alla preparazione ed all'esecuzione di una intera campagna di prove sperimentali.

3.1 – EQUAZIONI FONDAMENTALI

Le equazioni che descrivono il fenomeno fisico che si vuole analizzare sono generalmente espresse attraverso derivate ed integrali, e sono quindi risolvibili per via analitica solo quando il problema in esame è elementare, ovvero quando si considerano geometrie molto semplici e flussi laminari.

Nei problemi reali, invece, dove si hanno geometrie molto complesse e flussi turbolenti, la risoluzione del problema può avvenire soltanto per via numerica, con un processo iterativo di discretizzazione delle equazioni che viene ripetuto fino a convergenza.

L'accuratezza della soluzione ottenuta dipenderà quindi dal metodo di risoluzione scelto e dalla qualità del suo processo di discretizzazione.

Per effettuare un'analisi CFD occorre eseguire una modellazione matematica del problema fisico di interesse facendo riferimento ad equazioni di conservazione che vengono definite rispettando l'ipotesi di continuità, ovvero si considera che *“il fluido si distribuisca uniformemente all'interno del volume occupando interamente tutto lo spazio, e le sue proprietà fisiche e chimiche possano essere considerate come funzioni continue dello spazio e del tempo”*.

Grazie all'ipotesi di continuità, la conservazione di una generica grandezza $\phi(x, t)$ funzione dello spazio e del tempo in un volume di controllo Ω può essere espressa con l'equazione

$$\frac{d}{dt} \int_{\Omega} \rho \phi(x, t) d\Omega = \int_{\Omega} Q_{\Omega} d\Omega \quad (3.1)$$

in cui:

- ρ è la densità del fluido, espressa in $\left[\frac{kg}{m^3}\right]$
- x è una generica coordinata spaziale
- Q_{Ω} è il termine sorgente all'interno del volume Ω

Il termine alla sinistra dell'uguale nell'equazione (3.1) viene definito *derivata materiale* della grandezza ϕ e ne rappresenta la variazione complessiva nel tempo che, sviluppata, può essere scritta come

$$\frac{d}{dt} \int_{\Omega} \rho \phi(x, t) d\Omega = \frac{\partial}{\partial t} \int_{\Omega} \rho \phi(x, t) d\Omega + \int_S \rho \phi(x, t) \mathbf{U} \cdot \mathbf{n} dS \quad (3.2)$$

dove:

- \mathbf{U} è il vettore velocità del fluido in $\left[\frac{m}{s}\right]$
- \mathbf{n} è il versore normale alla superficie S di contorno del volume Ω

Utilizzando lo sviluppo della derivata materiale (3.2), l'equazione (3.1) diventa

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \phi(x, t) d\Omega + \int_S \rho \phi(x, t) \mathbf{U} \cdot \mathbf{n} dS = \int_{\Omega} Q_{\Omega} d\Omega \quad (3.3)$$

e, sostituendo il termine Q_{Ω} con il termine sorgente Q_{ϕ} della grandezza ϕ , si ottiene

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \phi(x, t) d\Omega + \int_S \rho \phi(x, t) \mathbf{U} \cdot \mathbf{n} dS = \int_{\Omega} Q_{\phi} d\Omega \quad (3.4)$$

che indica che la somma tra la derivata prima nel tempo ed il termine convettivo deve essere uguale alla variazione che la grandezza ϕ subisce all'interno del volume di controllo.

Usando il teorema di Gauss, l'integrale di superficie delle grandezza ϕ può essere scritto come l'integrale sul volume della sua divergenza, ovvero

$$\int_S \rho \phi(x, t) \mathbf{U} \cdot \mathbf{n} dS = \int_{\Omega} \nabla \cdot [\rho \phi(x, t) \mathbf{U}] d\Omega \quad (3.5)$$

e quindi, usando la relazione (3.5), l'equazione (3.4) può essere riscritta come

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \phi(x, t) d\Omega + \int_{\Omega} \nabla \cdot [\rho \phi(x, t) \mathbf{U}] d\Omega = \int_{\Omega} Q_{\phi} d\Omega \quad (3.6)$$

Se il volume di controllo Ω non varia nel tempo, l'equazione di bilancio (3.6) può essere scritta più semplicemente nella sua forma indefinita che esprime le derivate parziali senza fare riferimento ad un particolare volume di controllo

$$\frac{\partial}{\partial t} (\rho \phi) + \nabla \cdot (\rho \phi \mathbf{U}) = Q_{\phi} \quad (3.7)$$

Nei problemi fluidodinamici, partendo da questa relazione che è valida in generale per tutti i casi analizzati, si riesce a determinare il comportamento del sistema senza la necessità di definire una legge di moto, ma considerando solamente tre equazioni fondamentali che devono essere soddisfatte in tutta la regione di interesse al problema: la conservazione della massa, la conservazione della quantità di moto e la conservazione dell'energia.

3.1.1 – CONSERVAZIONE DELLA MASSA

Per un flusso comprimibile l'equazione di conservazione della massa, detta anche equazione di continuità, viene ricavata dalla relazione (3.7) imponendo $\phi = 1$ e $Q_\phi = 0$ perché non esiste termine sorgente per la massa, ovvero

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (3.8)$$

ovvero la somma tra la variazione della densità nel tempo ed il flusso di massa attraverso la superficie del volume di controllo deve essere nulla.

Se il flusso considerato è incomprimibile, ovvero se la densità ρ è costante, l'equazione di continuità (3.8) si semplifica e diventa solamente

$$\nabla \cdot \mathbf{U} = 0 \quad (3.9)$$

Considerando un sistema di coordinate cartesiane (x, y, z) e proiettando l'equazione (3.8) nelle tre direzioni, si ottiene

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho u_x) + \frac{\partial}{\partial y} (\rho u_y) + \frac{\partial}{\partial z} (\rho u_z) = 0 \quad (3.10)$$

dove le quantità u_x , u_y e u_z rappresentano le componenti del vettore velocità \mathbf{U} nelle tre coordinate.

3.1.2 – CONSERVAZIONE DELLA QUANTITÀ DI MOTO

La conservazione della quantità di moto per un flusso comprimibile può essere scritta come

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{U} d\Omega + \int_S \rho \mathbf{U} \mathbf{U} \cdot \mathbf{n} dS = \sum \mathbf{F} \quad (3.11)$$

in cui il vettore \mathbf{F} esprime le forze che agisce sul fluido ed è l'insieme delle forze di superficie (pressione, sforzi, tensione superficiale) e delle forze di volume (forza peso, forza centrifuga, forza elettromagnetica), ovvero

$$\sum \mathbf{F} = \int_S \bar{\boldsymbol{\sigma}} \cdot \mathbf{n} dS + \int_{\Omega} \rho \mathbf{b} d\Omega \quad (3.12)$$

in cui:

- $\bar{\boldsymbol{\sigma}}$ è il tensore degli sforzi
- \mathbf{b} è il vettore delle forze di volume

Sostituendo l'espressione (3.12) del vettore \mathbf{F} nell'equazione (3.11) di partenza si ottiene la relazione

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{U} d\Omega + \int_S \rho \mathbf{U} \mathbf{U} \cdot \mathbf{n} dS = \int_S \bar{\boldsymbol{\sigma}} \cdot \mathbf{n} dS + \int_{\Omega} \rho \mathbf{b} d\Omega \quad (3.13)$$

che, sfruttando ancora una volta il teorema di Gauss e ipotizzando che il volume di controllo Ω sia costante nel tempo, si può scrivere semplicemente come

$$\frac{\partial}{\partial t} (\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = \nabla \cdot \bar{\boldsymbol{\sigma}} + \rho \mathbf{b} \quad (3.14)$$

Il tensore degli sforzi $\bar{\boldsymbol{\sigma}}$ è una matrice che contiene informazioni sulla pressione e sugli effetti viscosi, e la sua generica componente σ_{ij} può essere scritta come

$$\sigma_{ij} = (-p + 2\lambda \nabla \cdot \mathbf{U}) \delta_{ij} + \tau_{ij} \quad (3.15)$$

dove:

- $\lambda = \left(\beta - \frac{2}{3}\mu\right)$ è un parametro funzione della viscosità di volume β e della viscosità dinamica μ
- δ_{ij} è detto *Delta di Kronecker* e vale 1 se $i = j$ e 0 se $i \neq j$
- τ_{ij} è la generica componente del tensore degli sforzi viscosi che, considerando l'ipotesi di *fluido newtoniano*, può essere espressa come

$$\tau_{ij} = 2\mu S_{ij} + \lambda S_{kk} \delta_{ij} \quad (3.16)$$

in cui la quantità S_{ij} è definita come

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3.17)$$

La componente idrostatica del tensore degli sforzi, ovvero il generico elemento diagonale della matrice, può essere scritto nella forma

$$\frac{1}{3} \sigma_{kk} = -p + \lambda S_{kk} + \frac{2}{3} \mu S_{kk} \quad (3.18)$$

Se il flusso è incomprimibile, la componente idrostatica σ_{kk} è semplicemente pari alla pressione p ; se invece il flusso è comprimibile, in genere si può assumere valida l'*ipotesi di Schlichting*

$$\lambda + \frac{2}{3} \mu = 0 \quad (3.19)$$

in modo che la viscosità compaia solamente nella parte deviatorica del tensore $\bar{\sigma}$ e la componente idrostatica sia ancora uguale soltanto alla pressione p , ovvero si verifica la cosiddetta *ipotesi di Stokes*.

Grazie a tutte queste considerazioni, la conservazione della quantità di moto (3.13) può essere scritta nella sua forma finale come

$$\frac{\partial}{\partial t} (\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot (\mu \mathcal{S}) + \rho \mathbf{b} \quad (3.20)$$

3.1.3 – CONSERVAZIONE DELL'ENERGIA

L'energia posseduta da un fluido viene facilmente espressa attraverso la sua entalpia h . Per arrivare alla scrittura di questa relazione si considera l'entalpia specifica del fluido

$$h = u + \frac{p}{\rho} \quad (3.21)$$

e la *Legge di Fourier* di trasferimento del calore per conduzione

$$\mathbf{q} = -k \nabla T \quad (3.22)$$

in cui $k = \frac{\mu C_p}{Pr}$ è la diffusività termica del fluido che dipende dalla viscosità dinamica μ , dal calore specifico C_p e dal numero di Prandtl Pr

La conservazione dell'energia viene scritta come

$$\frac{\partial}{\partial t} (\rho h) + \nabla \cdot (\rho h \mathbf{U}) = \nabla \cdot (k \nabla T) + \mathbf{U} \cdot \nabla p + \bar{\boldsymbol{\sigma}} \cdot \nabla \mathbf{U} + \frac{\partial p}{\partial t} \quad (3.23)$$

dove:

- $\nabla \cdot (k \nabla T)$ è il trasferimento di calore per conduzione secondo la legge di Fourier
- $\mathbf{U} \cdot \nabla p + \frac{\partial p}{\partial t}$ è il lavoro compiuto dalle forze di pressione
- $\bar{\boldsymbol{\sigma}} \cdot \nabla \mathbf{U}$ è il lavoro compiuto dalla componente idrostatica e dagli sforzi viscosi

Se il flusso è incomprimibile ed ha calore specifico costante, nella conservazione dell'energia (3.21) si può considerare direttamente la variazione di temperatura T invece dell'entalpia h

$$\frac{\partial}{\partial t} (\rho T) + \nabla \cdot (\rho T \mathbf{U}) = \nabla \cdot \left(\frac{\mu}{Pr} \nabla T \right) \quad (3.24)$$

dove $Pr = \frac{\mu C_p}{k}$ è il numero di Prandtl che dipende dalla viscosità dinamica μ , dal calore specifico C_p e dalla conducibilità termica k .

3.1.4 – EQUAZIONE DI STATO

Nel caso di flusso comprimibile è necessario considerare anche l'equazione di stato in modo da avere una relazione che permette di legare tra loro la pressione p , la temperatura T e la densità ρ .

$$\frac{p}{\rho} = R T \quad (3.25)$$

Unitamente a queste equazioni, si considerano delle ipotesi semplificative che permettono di ridurre la complessità del problema e quindi agevolarne la risoluzione. Queste semplificazioni possono riguardare:

- La geometria che rappresenta il problema.
- Le condizioni al contorno, ovvero i vincoli che le equazioni di conservazione devono rispettare.
- Il comportamento del fluido, che può essere trattato in modo semplificato trascurando gli effetti della viscosità e della comprimibilità.

Una volta definite le ipotesi semplificative appropriate per il problema in esame, prima di effettuare la vera e propria analisi CFD si scelgono i modelli da utilizzare, ovvero dei sistemi di equazioni che hanno lo scopo di descrivere gli effetti di un fenomeno fisico complesso come, ad esempio, lo scambio termico e la turbolenza, in modo da ridurre l'onere computazionale richiesto dall'analisi CFD.

La maggior parte di questi modelli è di natura semi-empirica, ovvero esprimono la variazione delle grandezze fluidodinamiche attraverso costanti determinate con prove sperimentali.

A questo punto, dopo aver definito i modelli da utilizzare, è possibile iniziare l'analisi CFD in cui, nella maggior parte dei casi, si utilizza il metodo dei volumi finiti per effettuare la discretizzazione del dominio di calcolo e delle equazioni.

3.2 – METODO DEI VOLUMI FINITI

Il punto di forza del metodo dei volumi finiti è l'integrazione delle equazioni di conservazione in un volume di controllo tridimensionale in modo da ottenere una chiara relazione tra l'algoritmo risolutivo e le leggi fisiche che governano il moto del fluido.

L'architettura del software *OpenFOAM* permette di risolvere le equazioni programmando espressioni che risultano molto simili alle formule matematiche reali.

Ad esempio, la risoluzione dell'equazione di conservazione della massa (3.8) può essere implementata come

```

{
    solve (fvm::ddt(rho) + fvc::div(phi));
}
    
```

Considerando l'equazione (3.7) di conservazione della generica grandezza ϕ ricavata in precedenza, con il termine sorgente Q_ϕ nullo, con il metodo dei volumi finiti questa viene integrata nel volume di controllo Ω per un intervallo di tempo Δt

$$\int_t^{t+\Delta t} \frac{\partial}{\partial t} \left(\int_\Omega \rho \phi(x, t) d\Omega \right) dt + \int_t^{t+\Delta t} \left(\int_\Omega \nabla \cdot [\rho \phi(x, t) \mathbf{U}] d\Omega \right) dt = 0 \quad (3.26)$$

e, considerando ancora una volta in teorema di Gauss, si può scrivere

$$\int_t^{t+\Delta t} \frac{\partial}{\partial t} \left(\int_\Omega \rho \phi(x, t) d\Omega \right) dt + \int_t^{t+\Delta t} \left(\int_S [\rho \phi(x, t) \mathbf{U}] \cdot \mathbf{n} dS \right) dt = 0 \quad (3.27)$$

Il passo successivo è quello di rendere discreta questa relazione integrale, ovvero trasformarla in una relazione algebrica più facilmente risolvibile. Questa conversione viene effettuata attraverso una serie di approssimazioni che prevedono la discretizzazione del dominio di calcolo e delle equazioni.

3.2.1 – DISCRETIZZAZIONE DEL DOMINIO

Discretizzazione spaziale

Per poter risolvere numericamente il problema, il dominio viene diviso in un numero finito di volumi di controllo più piccoli, detti *celle*, e prende così il nome di *griglia computazionale* o, più comunemente, *Mesh*.

La *Mesh* viene creata in modo tale che le celle di cui è costituita non si sovrappongano tra di loro e siano caratterizzate da un numero qualsiasi di facce piane orientate, in genere, secondo gli assi del sistema di riferimento scelto.

Nella griglia di calcolo così ottenuta tutte le variabili di interesse al problema (pressione, velocità, temperatura, ecc.) vengono associate al centro di ogni cella.

I *nodi di calcolo*, ovvero i punti in cui vengono discretizzate le equazioni, coincidono con i centroidi delle celle (P) in modo che valga la relazione

$$\int_{\Omega_P} (x - x_P) d\Omega = 0 \quad (3.28)$$

in cui:

- Ω_P è il volume della cella
- $(x - x_P)$ è la distanza tra la generica coordinata x ed il centro della cella x_P

Se si ipotizza che il valore della generica grandezza ϕ vari linearmente all'interno di ogni cella, considerando il valore ϕ_P che questa grandezza assume nel centroide, si può scrivere

$$\phi = \phi_P + (x - x_P)(\nabla\phi)_P + o((x - x_P)^2) \quad (3.29)$$

e quindi il generico integrale di volume può essere discretizzato nel dominio come

$$\int_{\Omega} \phi d\Omega = \int_{\Omega} \phi_P d\Omega + \int_{\Omega} (x - x_P)(\nabla\phi)_P d\Omega + \int_{\Omega} o(|x - x_P|^2) d\Omega \quad (3.30)$$

$$\int_{\Omega} \phi d\Omega = \phi_P \Omega_P + o(|\Delta x|^2) \quad (3.31)$$

ovvero l'integrale di volume della grandezza ϕ può essere approssimato con il prodotto tra il valore ϕ_P della grandezza nel centroide (P) ed il volume della cella Ω_P , ottenendo un'accuratezza del secondo ordine.

Esistono principalmente due tipologie di *Mesh* con cui può essere discretizzato il dominio, ognuna con caratteristiche differenti.

Mesh Structured

In una *Mesh* di tipo *Structured* le celle che compongono la griglia sono connesse tra di loro in modo molto regolare; questo consente di dividere in modo molto efficiente il volume di controllo con benefici in termini di convergenza e precisione della soluzione.

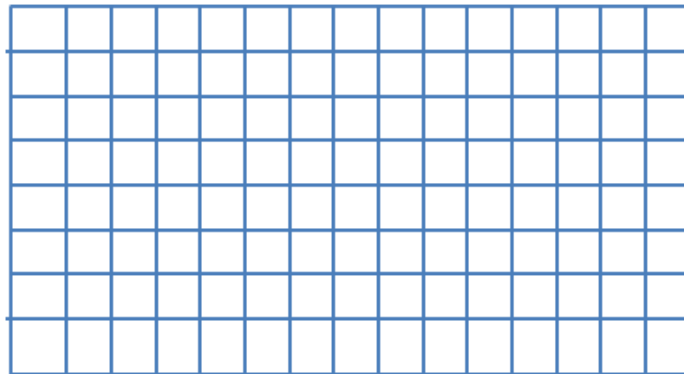


Figura 3.1 – Esempio di *Mesh Structured*

Mesh Unstructured

Le celle di una *Mesh* di tipo *Unstructured* sono collegate tra di loro in modo irregolare; questo comporta una riduzione dell'efficienza della griglia ma, al tempo stesso, una maggiore flessibilità che permette un migliore adattamento a geometrie complesse.

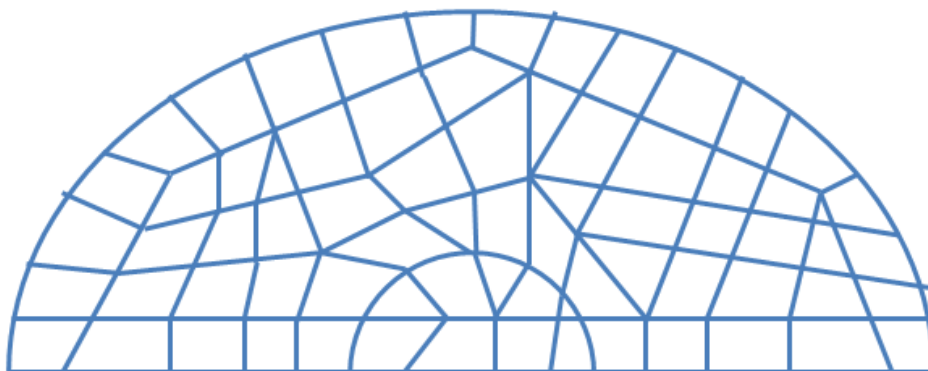


Figura 3.2 – Esempio di *Mesh Unstructured*

In alcuni casi è utile fare ricorso a *Mesh ibride*, che prevedono di utilizzare porzioni di griglia di tipo *Structured* nelle zone in cui la geometria è semplice al fine di semplificare il processo di risoluzione, e porzioni *Unstructured* nelle zone dove la geometria è più complessa.

La scelta della tipologia e del numero di elementi che compongono la *Mesh* deve essere eseguita con cura perché da queste grandezze dipendono direttamente le capacità del modello di descrivere correttamente il fenomeno fisico ed i tempi di calcolo necessari alla risoluzione del problema.

In tutti i nodi della griglia di calcolo vengono risolti, con un metodo iterativo, tutti i sistemi di equazioni determinati in precedenza grazie ai modelli.

Discretizzazione temporale

Unitamente alla discretizzazione spaziale, il dominio subisce anche un discretizzazione temporale, ovvero viene diviso in una serie di passi temporali Δt che possono variare durante l'esecuzione della simulazione in modo da rispettare particolari condizioni di calcolo.

In particolare, nella fluidodinamica computazionale è necessario rispettare la condizione di *Courant-Friedrichs-Lewy* secondo la quale, “*se un’onda attraversa una griglia di calcolo discreta, allora l’intervallo temporale deve essere più piccolo del tempo necessario all’onda per attraversare due punti adiacenti della griglia*”.

Per rispettare questa condizione, necessaria per la convergenza della soluzione delle equazioni, deve valere la relazione

$$\frac{U \Delta t}{\Delta x} < C \quad (3.32)$$

dove:

- U è la velocità
- Δt è l’intervallo temporale
- Δx è la spaziatura della griglia di calcolo
- C è una costante detta *numero di Courant* che, per garantire la corretta convergenza della soluzione, deve essere inferiore ad 1.

3.2.2 – DISCRETIZZAZIONE DELLE EQUAZIONI

Discretizzare le equazioni significa trasformarle in equazioni algebriche, ovvero passare da una relazione formata da una successione di integrali ad una relazione composta soltanto da sommatorie.

Considerando, come spiegato nel paragrafo precedente, che tutte le grandezze siano riferite al centroide di ogni cella, per un flusso instazionario la formulazione generale dell'equazione del trasporto per un generica grandezza ϕ assume la forma

$$\frac{\partial}{\partial t}(\rho \phi) + \nabla \cdot (\rho \phi \mathbf{U}) = \nabla \cdot (\Gamma_\phi \nabla \phi) + S_\phi(\phi) \quad (3.33)$$

che, integrata sul volume di controllo Ω , diventa

$$\begin{aligned} \int_{\Omega} \frac{\partial}{\partial t}(\rho \phi) d\Omega + \int_{\Omega} \nabla \cdot (\rho \phi \mathbf{U}) d\Omega = \\ = \int_{\Omega} \nabla \cdot (\Gamma_\phi \nabla \phi) d\Omega + \int_{\Omega} S_\phi(\phi) d\Omega \end{aligned} \quad (3.34)$$

in cui:

- $\frac{\partial}{\partial t}(\rho \phi)$ è la derivata temporale della grandezza ϕ
- $\nabla \cdot (\rho \phi \mathbf{U})$ è il termine convettivo
- $\nabla \cdot (\Gamma_\phi \nabla \phi)$ è il termine diffusivo
- $S_\phi(\phi)$ è il termine sorgente

Discretizzazione della derivata temporale

La derivata temporale viene in genere approssimata usando delle relazioni alle differenze finite. Si possono utilizzare schemi diversi che permettono di ottenere differenti gradi di accuratezza.

Metodo di Eulero

È lo schema più semplice, con un'accuratezza del primo ordine, con cui si approssima la derivata con la relazione

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi^n - \phi^{n-1}}{\Delta t} \quad (3.35)$$

Metodo di Crank – Nicholson

È un'approssimazione più complessa che permette di ottenere un'accuratezza del secondo ordine.

$$\frac{\partial \phi}{\partial t} \approx (1 + \alpha) \left[\frac{\phi^n - \phi^{n-1}}{\Delta t} - \alpha (1 + \alpha) \frac{\phi^{n-1} - \phi^{n-2}}{\Delta t} - \alpha \phi^{n-2} \right] \quad (3.36)$$

in cui α può assumere valori compresi tra 0 e 1.

Discretizzazione del termine convettivo

Il termine convettivo viene discretizzato nel modo seguente

$$\int_{\Omega} \nabla \cdot (\rho \phi \mathbf{U}) d\Omega = \int_S \mathbf{n} \cdot (\rho \phi \mathbf{U}) dS = \sum_f S_f \cdot (\rho \mathbf{U})_f \phi_f \quad (3.37)$$

dove:

- $S_f \cdot (\rho \mathbf{U})_f$ è il flusso volumetrico che attraversa una faccia della cella e viene determinato con algoritmi specifici
- ϕ_f è una variabile incognita che rappresenta il valore della grandezza ϕ nel centro delle facce del volume di controllo e può essere determinato usando metodi diversi

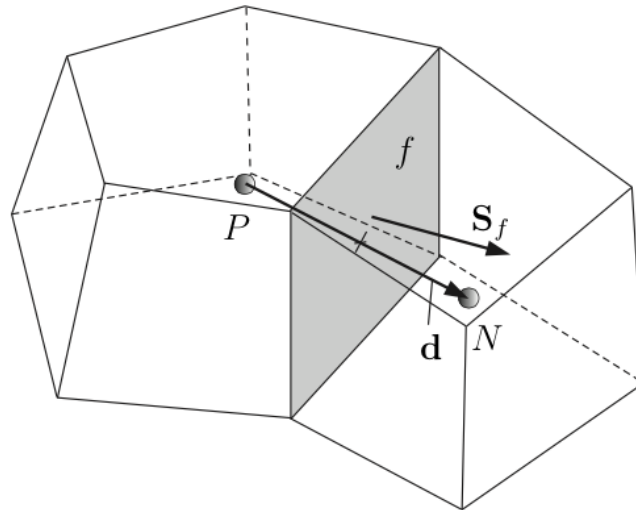


Figura 3.3 – Dettaglio della faccia f in comune a due celle adiacenti

Nella relazione (3.37) la variabile ϕ_f è incognita perché, durante il processo di discretizzazione, i valori delle grandezze di interesse vengono calcolati nel centroide delle celle e non sulle loro facce.

Per determinare il valore di ϕ_f bisogna quindi ricorrere a tecniche di interpolazione che devono essere scelte con cura perché da esse dipende direttamente il grado di accuratezza del modello.

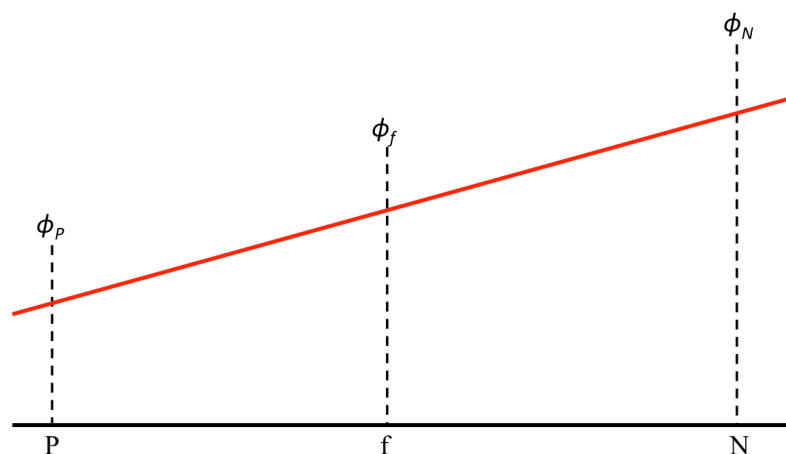


Figura 3.4 – Interpolazione del valore ϕ_f sulla faccia f

Considerando due celle adiacenti di centroidi (P) ed (N), in cui il valore della grandezza ϕ è pari rispettivamente a ϕ_P e ϕ_N , il valore ϕ_f sulla faccia comune f può essere determinato in tre modi differenti.

Central Differencing (CD)

Si ipotizza un andamento lineare della grandezza ϕ lungo il segmento \overline{PN} congiungente i due centroidi, e si può quindi scrivere

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N \quad (3.38)$$

dove $f_x = \frac{\overline{fN}}{\overline{PN}}$

Questa interpolazione permette di ottenere una accuratezza del secondo ordine anche per griglie di calcolo non uniformi.

Upwind Differencing (UD)

Il valore della grandezza ϕ_f viene approssimato considerando soltanto la direzione del flusso F , ottenendo un'accuratezza solo del primo ordine.

$$\phi_f = \begin{cases} \phi_P & \text{se } F \geq 0 \\ \phi_N & \text{se } F < 0 \end{cases} \quad (3.39)$$

Blended Differencing (BD)

Si combinano linearmente le due tecniche precedenti per ottenere la relazione

$$\phi_f = (1 - \gamma) (\phi_f)_{UD} + \gamma (\phi_f)_{CD} \quad (3.40)$$

dove γ è detto *blending factor* e può assumere valori compresi tra 0 e 1.

Discretizzazione del termine diffusivo

Il termine diffusivo viene discretizzato in modo simile al termine convettivo, assumendo la forma

$$\begin{aligned} \int_{\Omega} \nabla \cdot (\Gamma_{\phi} \nabla \phi) d\Omega &= \int_S \Gamma_{\phi} \nabla \phi \cdot \mathbf{n} dS = \\ &= \sum_f S_f \cdot (\rho \Gamma_{\phi} \nabla \phi)_f = \sum_f (\rho \Gamma_{\phi})_f (S \cdot \nabla \phi)_f \end{aligned} \quad (3.41)$$

in cui:

- $(\rho \Gamma_{\phi})_f$ viene interpolato sulla faccia
- $(S \cdot \nabla \phi)_f$ dipende dal tipo di *Mesh*

CAPITOLO 4

CASO 3D

Per simulare il funzionamento del compressore scroll vengono dapprima creati i modelli 3D dei profili di statore e rotore utilizzando il software CAD Autodesk Inventor e, successivamente, usando le applicazioni ed i risolutori di *OpenFOAM* e della libreria *LibRotaryMachine* si riproduce il movimento della macchina.

4.1 – CREAZIONE DELLA GEOMETRIA: 1° METODO

Non avendo a disposizione parametri precisi sulla geometria della macchina, ci si basa su quanto disponibile in letteratura; i profili di statore e rotore vengono creati scegliendo delle dimensioni arbitrarie cercando di rispettare le proporzioni reali tra le varie parti del compressore.

Il profilo del rotore viene creato come una spirale di altezza pari a 38 millimetri, spessore 2 millimetri, avvolta su se stessa per tre volte, ovvero su un angolo di 1080°.

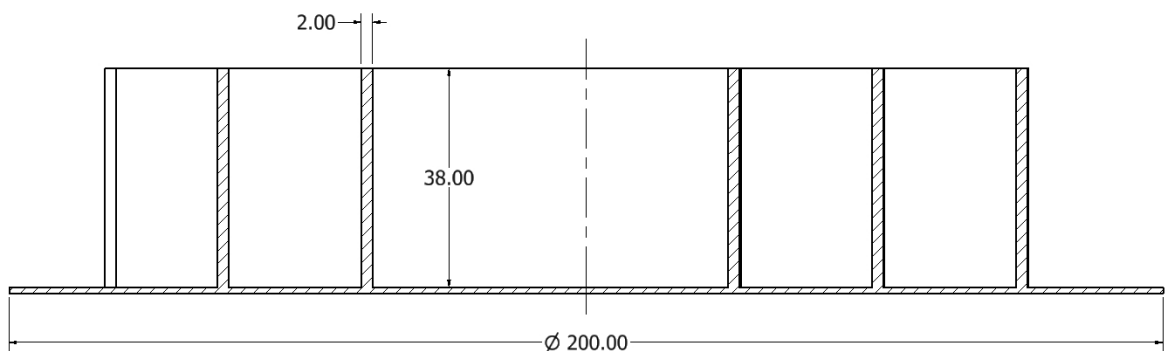


Figura 4.1 – Vista in sezione del rotore

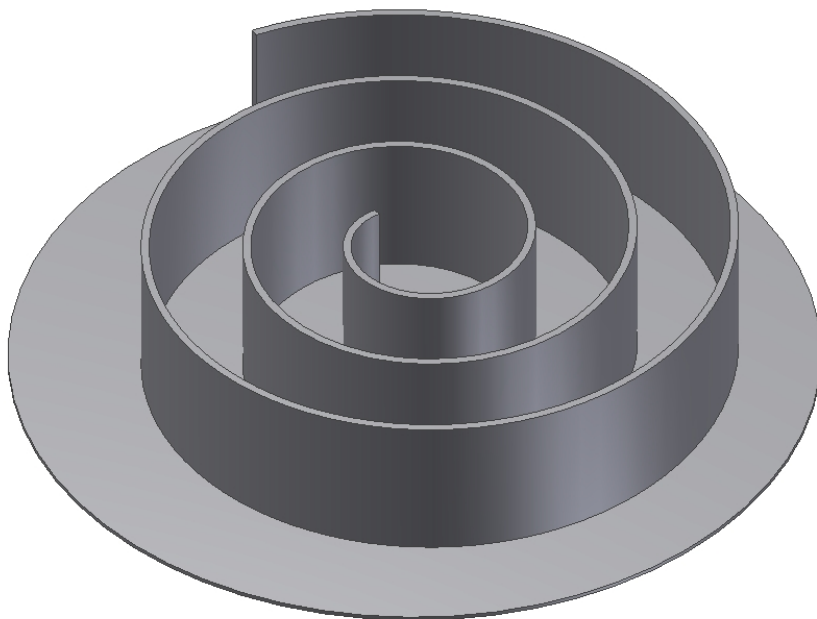


Figura 4.2 – Vista 3D del rotore

Il profilo dello statore è esattamente identico a quello del rotore e quindi può essere facilmente ottenuto ribaltando di 180° il profilo precedente per da avere una geometria speculare a quella del rotore.

A questo profilo si aggiunge anche una superficie cilindrica, concentrica alla spirale, al fine di simulare il condotto di scarico della macchina.

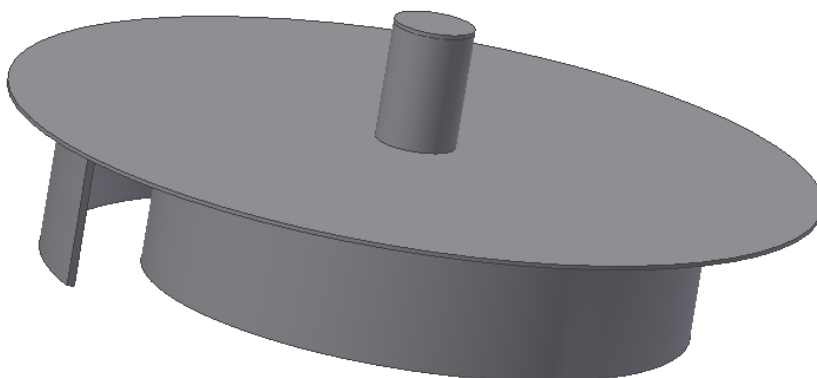


Figura 4.3 – Vista 3D dello statore

I modelli 3D così creati, per poter essere riconosciuti da *OpenFOAM*, vengono salvati in formato STL (Stereo Lithography Interface), ovvero un formato in cui la superficie del solido viene discretizzata in triangoli.

I file in formato STL sono molto comodi da usare perché risultano semplici da generare e da manipolare ma, a causa della discretizzazione della superficie del solido, questo tipo di file non permette di riprodurre fedelmente la geometria reale ma solo una sua approssimazione.

Con l'utilizzo di specifiche funzioni di *OpenFOAM* i profili 3D vengono scomposti in una serie di elementi, detti *patch*, che rendono facilmente distinguibili le varie componenti della geometria della macchina e permettono una più agevole applicazione delle condizioni al contorno.

Le *patch* in cui è stata divisa la geometria sono le seguenti:

- *top*: superficie superiore di chiusura delle spirali
- *bottom*: superficie inferiore di chiusura delle spirali
- *stator*: spirale dello statore
- *rotor*: spirale del rotore
- *inlet*: sezione di ingresso del fluido nella macchina
- *outlet*: sezione di uscita del fluido dalla macchina
- *duct*: condotto di scarico del fluido

I profili sono poi uniti posizionandoli uno dentro l'altro e, per simulare il funzionamento reale della macchina, viene definita come sezione di ingresso del fluido tutta la superficie circolare di contorno dei profili, e come scarico la parte superiore del condotto di scarico dello statore.

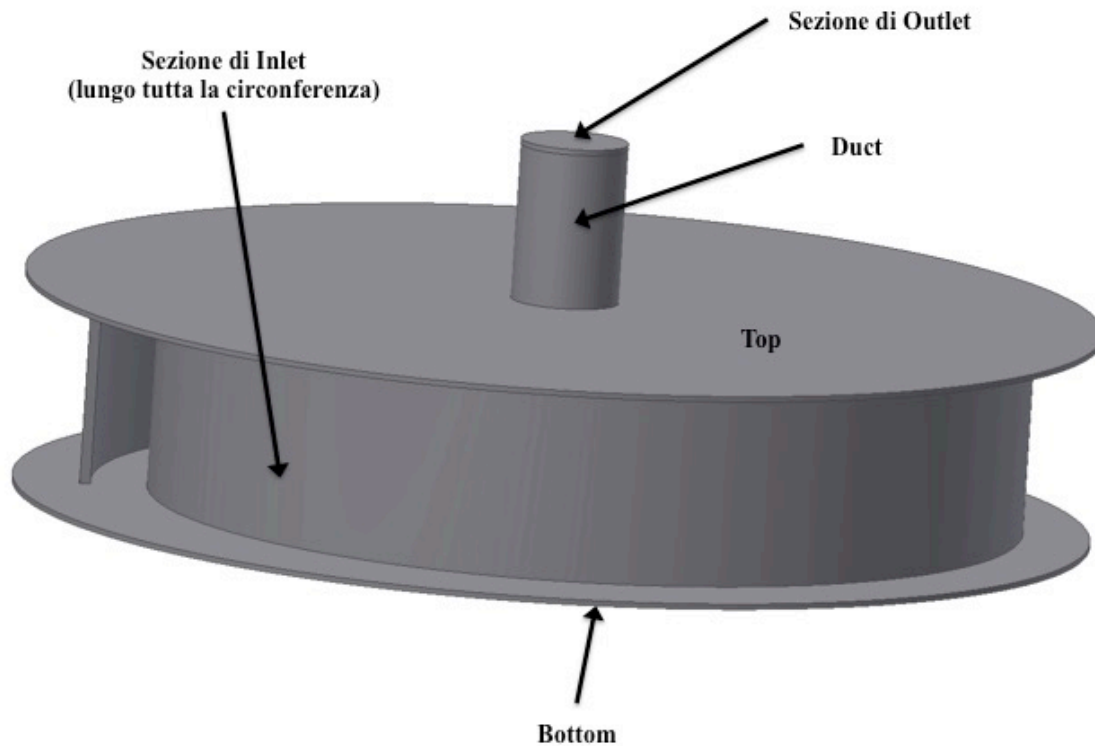


Figura 4.4 – Vista 3D del gruppo statore e rotore

Dopo aver creato la geometria 3D della macchina, il passo successivo è quello di generare la griglia di calcolo che racchiude i profili, ovvero il volume di controllo in cui verrà eseguita la simulazione.

Per fare questo si utilizza l'utility *BlockMesh* di *OpenFOAM* che, attraverso l'implementazione di un dizionario specifico, permette di definire la geometria della *Mesh* con un insieme di punti ognuno dei quali è caratterizzato da tre coordinate (x, y, z) che ne identificano la posizione nelle tre direzioni del sistema di riferimento.

Nello stesso dizionario è anche possibile specificare il numero di celle di cui deve essere composta la *Mesh*.

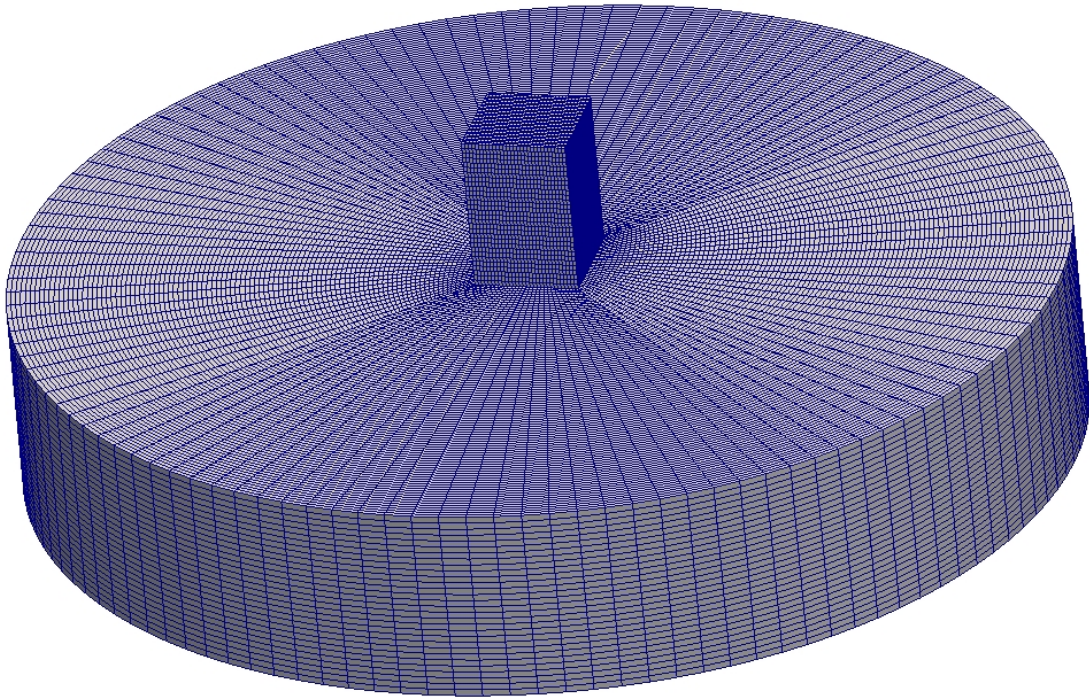


Figura 4.5 – Vista 3D della *Mesh* di calcolo con 415.000 celle

L'obiettivo della simulazione è quello di riprodurre il movimento del compressore su un giro completo, da 0° a 360° ; per fare questo il profilo del rotore deve seguire una traiettoria circolare, ovvero gli si impone una traslazione nelle due direzioni pari a

$$\begin{cases} dx = r * \cos(\vartheta) \\ dy = r * \sin(\vartheta) \end{cases} \quad (4.1)$$

dove:

- r è il raggio di curvatura della traiettoria circolare
- ϑ è l'angolo che viene fatto variare progressivamente tra 0° e 360°

Nella macchina reale tra lo statore ed il rotore sono presenti dei *gap* in direzione assiale, ovvero tra l'apice di una spirale e la base dell'altra, e radiale, tra i fianchi delle due spirali.

Queste distanze sono necessarie affinché durante il movimento i due profili non vengano mai in contatto tra loro consentendo così di limitarne l'usura, al tempo stesso, però, sono dannose perché comportano anche una riduzione delle prestazioni della macchina a causa del trafilamento del fluido.

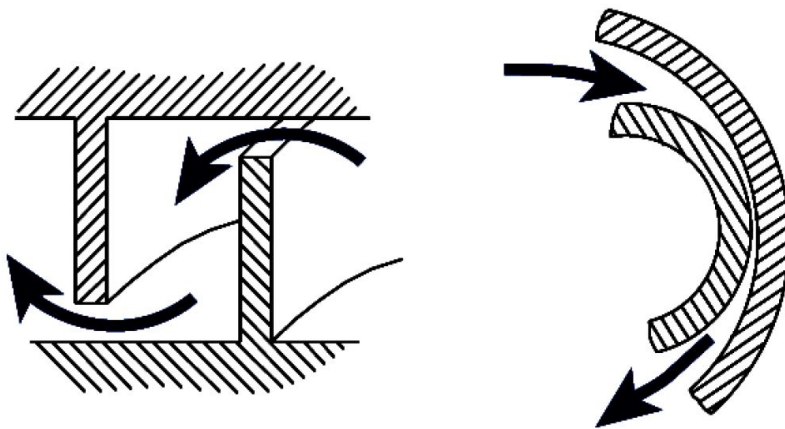


Figura 4.6 – Trafilamento attraverso il *gap* assiale (a sinistra) e radiale (a destra)

Per limitare le perdite dovute al trafilamento del fluido i *gap* sono in genere molto piccoli, intorno a 0,03 millimetri in direzione radiale e 0,025 millimetri in direzione assiale.

Durante la simulazione, per ridurre i tempi e gli oneri computazionali necessari alla generazione della *Mesh*, si sceglie una traiettoria circolare con un raggio di curvatura pari a $r = 0,0102$ metri che è in grado di assicurare, nella zona centrale della spirale, una distanza tra statore e rotore di 0,08 millimetri, valore superiore a quello reale ma sufficientemente basso da garantire una buona simulazione.

In questo modo si riesce a ridurre il numero di celle presenti nella griglia di calcolo e quindi a rendere più rapida l'esecuzione del codice.

Sempre per esigenze di riduzione dei tempi necessari alla simulazione, si considera il caso di “macchina sigillata”, ovvero una tipologia di compressore in cui è presente una guarnizione nella parte superiore delle spirali che impedisce il trafilamento del fluido e permette di fatto di annullare il *gap* assiale.



Figura 4.7 – Esempio di macchina sigillata

Per rendere ancora più rapido tutto il processo si effettuano delle ulteriori semplificazioni sulla griglia di calcolo che, invece di essere creata direttamente in forma tridimensionale, viene generata come se fosse soltanto 2D per essere poi riconvertita in 3D attraverso un processo di estrusione della geometria ottenuta.

In questo modo si ha una notevole riduzione del numero di celle di cui è composta la *Mesh* e quindi tempi computazionali più bassi.

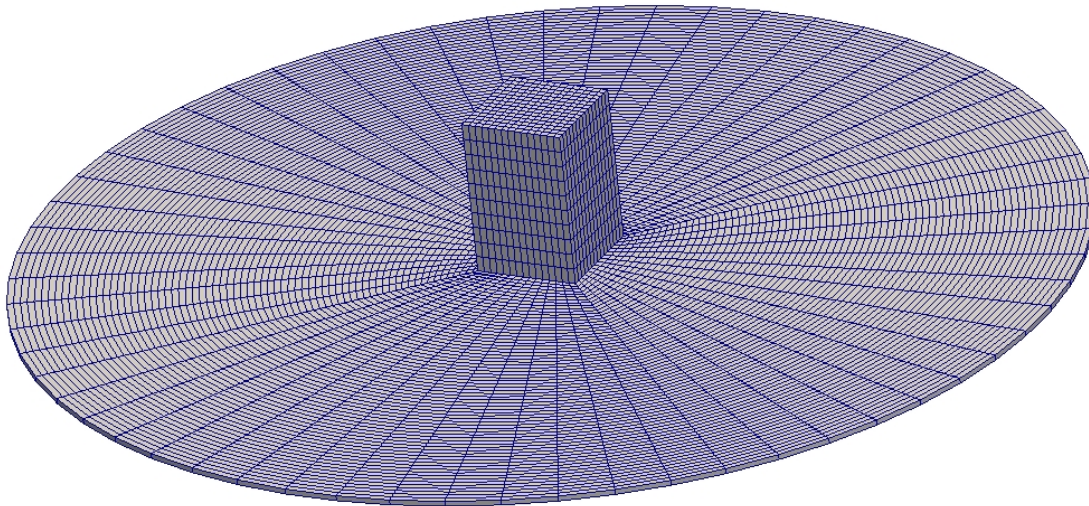


Figura 4.8 – Vista 3D della Mesh di calcolo semplificata con 6.800 celle

Dopo aver creato la griglia che contiene la geometria della macchina, usando l'utility *SnappyHexMesh* è possibile effettuare un adeguamento della griglia alle superfici di statore e rotore.

Grazie a questa utility, con un processo iterativo la griglia dalla forma semplice creata con il *BlockMesh* viene progressivamente trasformata per adattarsi alle superfici delle geometrie STL che vi sono contenute. Le celle che non appartengono al dominio di interesse vengono rimosse e quelle che si intersecano con la geometria da analizzare vengono via via rifinite fino ad arrivare ad ottenere una griglia composta da celle esaedriche e quindi di elevata qualità.

Dopo questo processo, definito *Snap*, grazie all'utility *ExtrudeMesh* si riporta la geometria alle condizioni tridimensionali originarie.

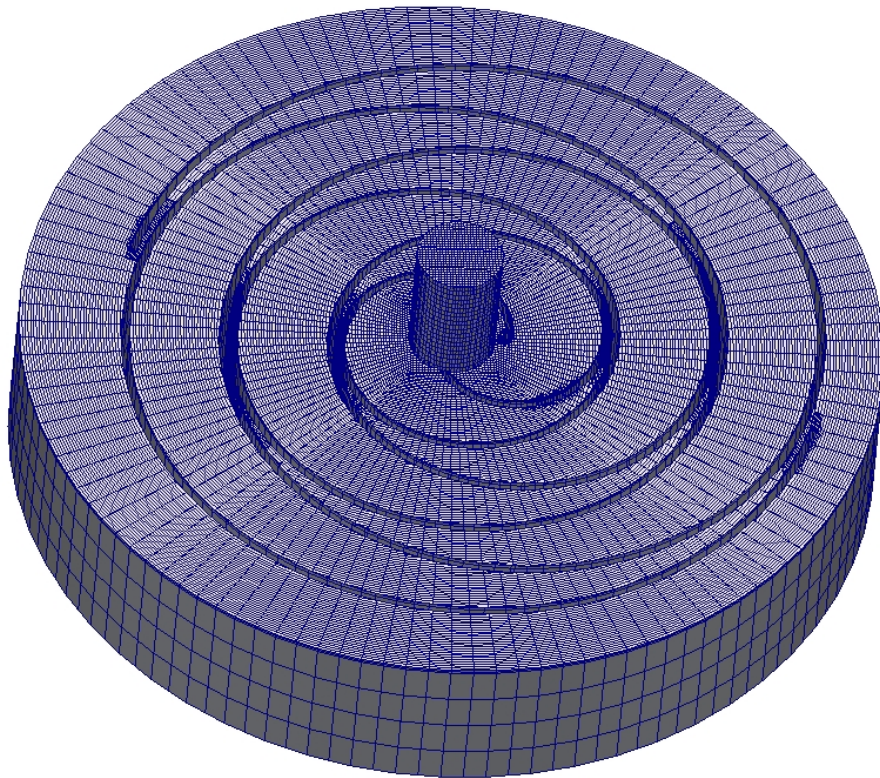


Figura 4.9 – Vista 3D della Mesh finale

L'utilizzo di questo metodo per la creazione della *Mesh* crea due problemi:

- Con un gap assiale nullo il profilo del rotore deve “tagliare” la superficie *top* dello statore.
- La superficie *top* coincide con la parte superiore della griglia creata con l'utility *BlockMesh*.

Questi problemi causano una imperfetta generazione della *Mesh* in corrispondenza degli spigoli dei profili, che comporta la formazione di celle distorte che possono causare inconvenienti di oscillazione dei risultati arrivando fino ad una mancata convergenza durante l'esecuzione del risolutore.

Per migliorare questo aspetto si genera la geometria 3D della macchina con un metodo alternativo.

4.2 – CREAZIONE DELLA GEOMETRIA: 2° METODO

Per evitare problemi di distorsione delle celle durante la generazione della *Mesh* la geometria del compressore descritta nel paragrafo precedente viene realizzata con una procedura alternativa che prevede di affidarsi maggiormente all'utility *BlockMesh* al fine di realizzare una griglia con celle di elevata qualità.

Questo processo inizia creando il condotto di scarico *duct* della macchina, chiuso all'estremità superiore dalla superficie di *outlet* del fluido

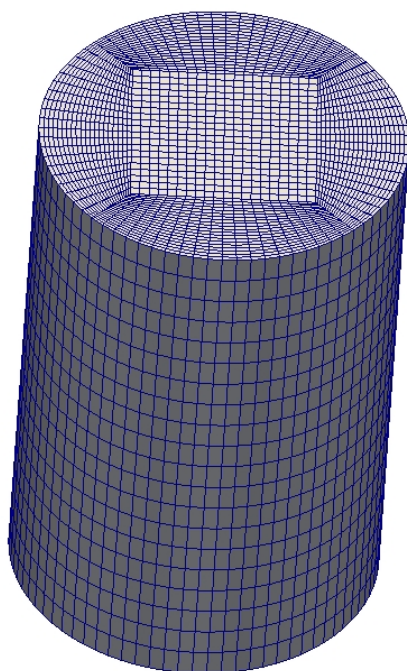


Figura 4.10 – Vista 3D del condotto di scarico realizzato con l'utility *BlockMesh*

A questa griglia cilindrica viene aggiunto, nella parte inferiore, un ulteriore livello di celle spesso 1 millimetro che sarà poi estruso radialmente per ottenere la superficie superiore *top*.

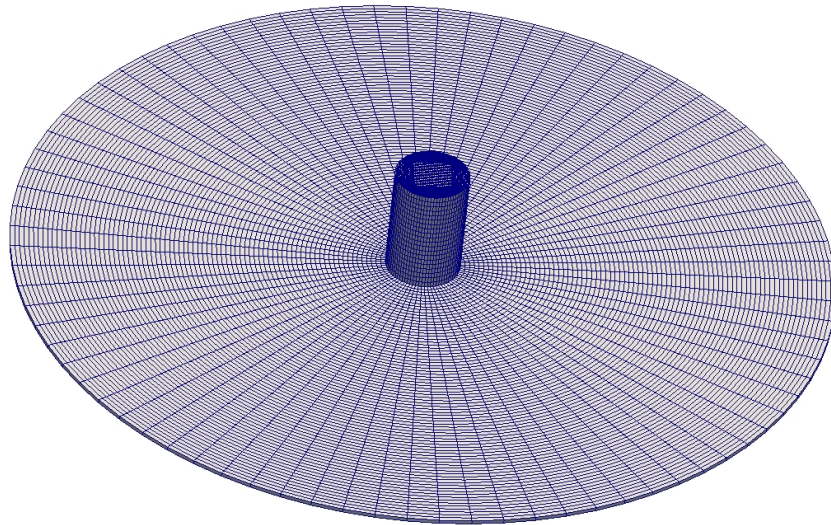


Figura 4.11 – Vista 3D dell'insieme *duct* e superficie *top*

A questa geometria vengono poi aggiunti i profili di statore e rotore, che in questo caso sono soltanto delle semplici spirali senza ulteriori superfici, e viene effettuato il processo di *Snap* della griglia.

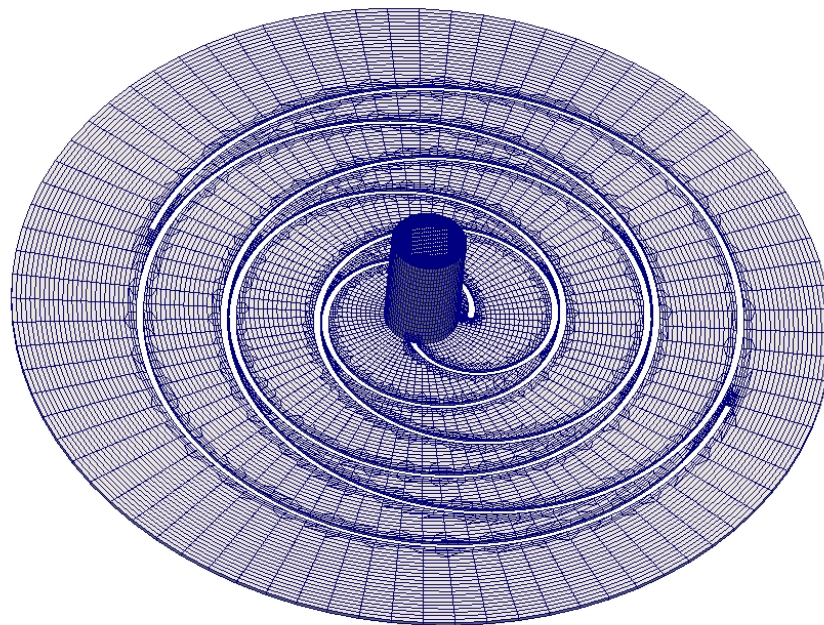


Figura 4.12 – Risultato del processo di *Snap*

Alla fine la *Mesh* viene estrusa in direzione assiale per conferirle l'altezza reale ed ottenere una geometria tridimensionale (Figura 4.9).

Questo metodo alternativo consente di ottenere una geometria del tutto identica a quella realizzata con il metodo precedente, con le stesse dimensioni e gli stessi valori di *gap* assiale e radiale; si ha però il grande vantaggio di riuscire a realizzare una *Mesh* di qualità superiore, con celle molto regolari e non distorte soprattutto in corrispondenza degli spigoli.

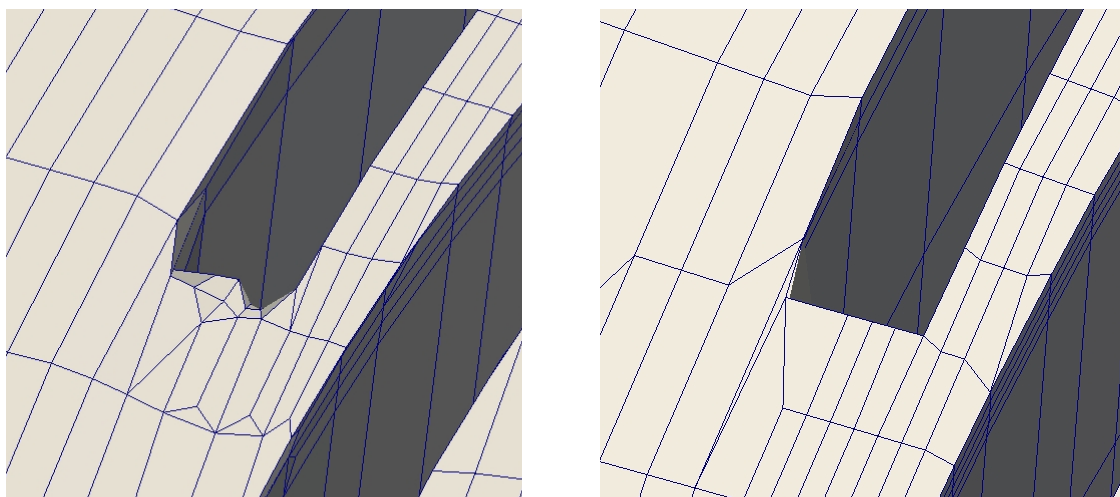


Figura 4.13 – Confronto tra la qualità delle celle in corrispondenza degli spigoli ottenute con il primo metodo (a sinistra) e con il secondo metodo (a destra)

4.3 – SIMULAZIONE DEL FUNZIONAMENTO

Tutto il processo di generazione della *Mesh* descritto nel paragrafo precedente viene automatizzato creando uno script chiamato *CreateBaseMesh* che, una volta eseguito, impone la traslazione al rotore ed esegue in successione le utility *BlockMesh*, *SnappyHexMesh* ed *ExtrudeMesh*.

Il funzionamento della macchina viene simulato utilizzando, oltre allo script *CreateBaseMesh*, due utility specificatamente sviluppate presenti nella libreria *LibRotaryMachine*.

4.3.1 – *rotaryMachineDynamicSetUp*

È una utility di pre-processing che crea dinamicamente la *Mesh* al variare dell'angolo di rotazione del rotore da 0° e 360° , testandone continuamente le caratteristiche ed interrompendo il movimento quando la qualità della *Mesh* diventa scadente.

A partire dall'angolo iniziale $\vartheta_0 = 0^\circ$, viene lanciato lo script *CreateBaseMesh* e si esegue il movimento del rotore fino ad arrivare ad un angolo $\vartheta_1 > \vartheta_0$ oltre il quale la qualità della *Mesh* peggiora. L'angolo ϑ_1 diventa quindi il nuovo angolo ϑ_0 di partenza e tutto il processo viene ripetuto iterativamente fino a coprire tutti i 360° di un giro completo.

La qualità della *Mesh* viene testata utilizzando due proprietà che vengono definite sulle facce delle celle che la compongono, e quindi non dipendono dalla forma del volume di controllo utilizzato.

Non-Ortogonalità

La non-ortogonalità analizza l'angolo tra le facce di due elementi adiacenti e lo confronta con il suo valore ottimale che deve essere pari a 90° per elementi con facce quadrate e 60° per elementi con facce triangolari. Maggiore è la differenza tra il valore ottimale ed il valore effettivo dell'angolo e peggiore è la qualità della *Mesh*.

Skewness

Lo skewness è un parametro che controlla la qualità della mesh sulla base dei suoi parametri geometrici.

Considerando la dimensione di una cella della griglia, il valore di skewness può essere determinato come

$$S = \frac{D_O - D_E}{D_O} \quad (4.2)$$

dove:

- D_O è la dimensione ottimale della cella
- D_E è la dimensione effettiva della cella

Il valore di skewness ricavato con questa relazione può variare tra 0 (qualità della *Mesh* elevata) e 1 (qualità della *Mesh* molto scadente).

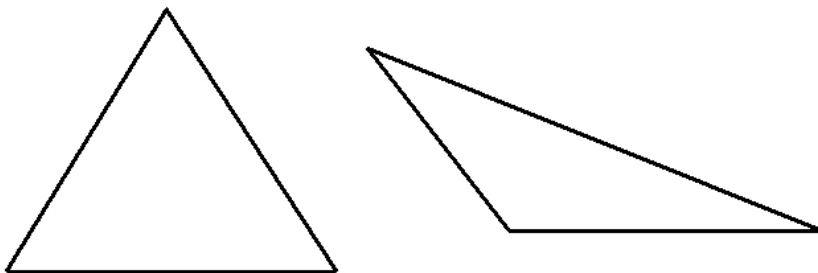


Figura 4.14 – Esempio di cella con *skewness* nullo (a sinistra) e con *skewness* elevato (a destra)

Con lo scopo di rendere più precisa la simulazione, oltre a considerare le caratteristiche di non-ortogonalità e skewness della griglia, è stato implementato anche un dizionario che impone un limite massimo di movimento al rotore.

In questo modo, anche se la qualità della *Mesh* è ancora buona, ogni 2° la rotazione viene interrotta e si passa alla generazione di una nuova *Mesh*.

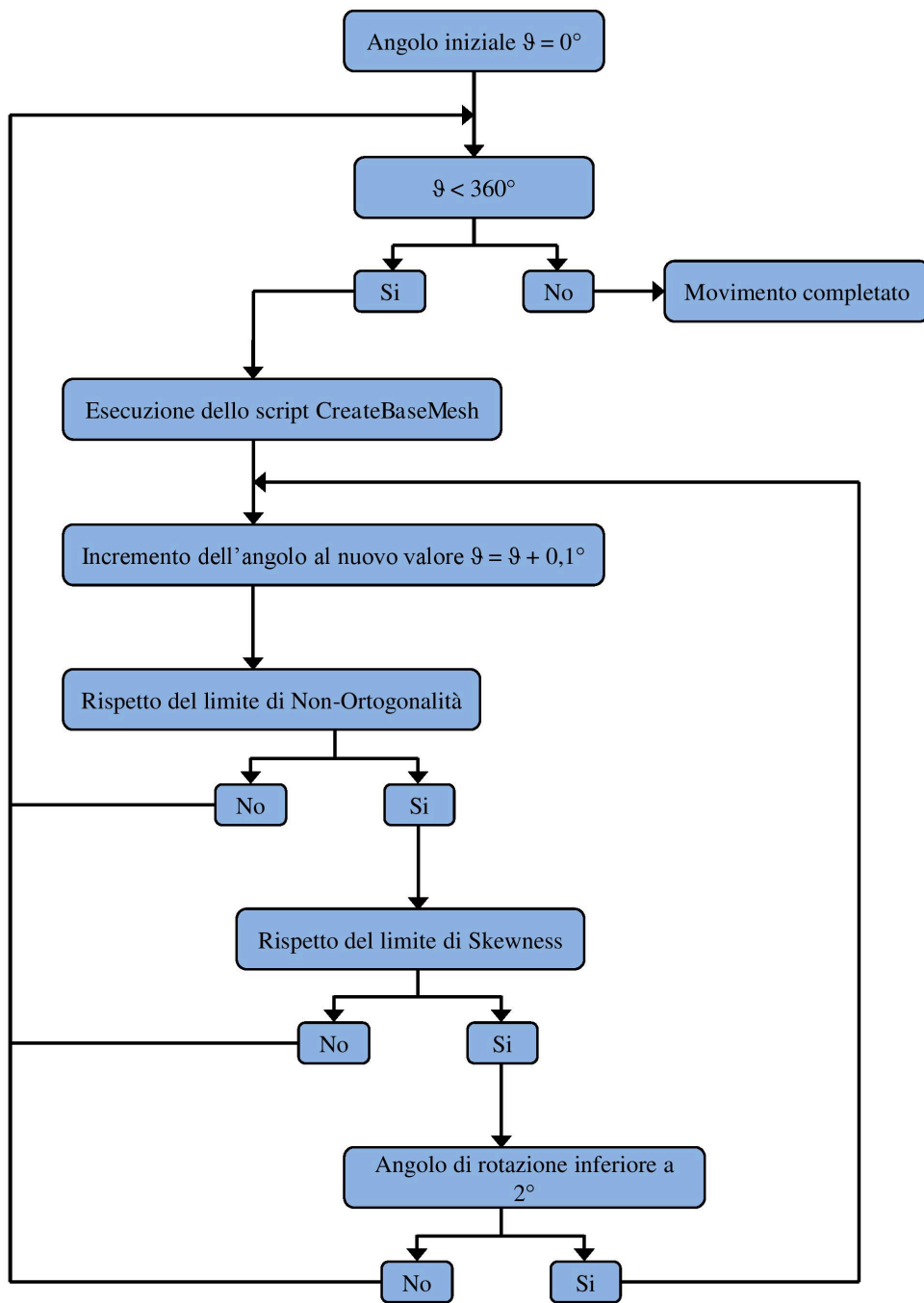


Figura 4.15 – Schema di funzionamento dell’utility *rotaryMachineDynamicSetUp*

4.3.2 – *runMultiMeshCase*

È una utility che permette di eseguire il problema in esame considerando tutte le *Mesh* generate dalla *rotaryMachineDynamicSetUp* unitamente alle caratteristiche fisiche e termodinamiche del fluido reale.

L'esecuzione di questa utility permette di risolvere il problema fluidodinamico legando tra loro tutte le *Mesh* generate in precedenza.

All'inizio della simulazione vengono considerati come dati iniziali le condizioni al contorno impostate dall'utente e, all'aumentare dell'angolo di rotazione del rotore, il risolutore *rotaryMachineFoam* calcola automaticamente le variazioni di pressione, velocità e temperatura del fluido all'interno della *Mesh*.

Ogni volta che si ha il passaggio da una *Mesh* alla successiva viene eseguito un rimappaggio dei campi di moto, ovvero, le condizioni raggiunte del fluido alla fine del movimento della *Mesh* precedente diventano le condizioni iniziali per la *Mesh* successiva.

Per evitare problemi durante la fase di rimappaggio dei valori di pressione, temperatura e velocità, si è scelto di eseguire questo processo appoggiandosi alla vecchia versione della libreria *LibRotaryMachine*. Il risolutore è stato quindi impostato per determinare la variazione dei campi di moto usando la *LibRotaryMachine* nella versione 2.3 per poi eseguire il rimappaggio con la versione 2.2 e infine ritornare alla 2.3 per riprendere i calcoli della nuova *Mesh*.

CAPITOLO 5

ANALISI DI SENSITIVITÀ AI GAP

Come specificato nel capitolo precedente, per ridurre gli oneri computazionali richiesti dalla simulazione viene considerato un *gap* tra statore e rotore di 0,08 millimetri in direzione radiale e nullo in direzione assiale.

L'analisi di sensitività ha lo scopo di determinare come e quanto i differenti valori di *gap* tra i profili influenzano il numero di celle presenti nella *Mesh* ed i tempi necessari alla sua realizzazione.

Per ottenere valori di *gap* il più possibile vicini a quelli reali occorre scegliere con cura il numero di celle che compongono il dominio di calcolo e stabilire opportuni livelli di *refinement*, ovvero un infittimento della griglia nelle zone in cui i due profili sono molto vicini tra loro.

Nella versione 2.3 del software *OpenFOAM* questo è possibile settando opportunamente il parametro *gapLevelIncrement* dell'utility *snappyHexMesh*.

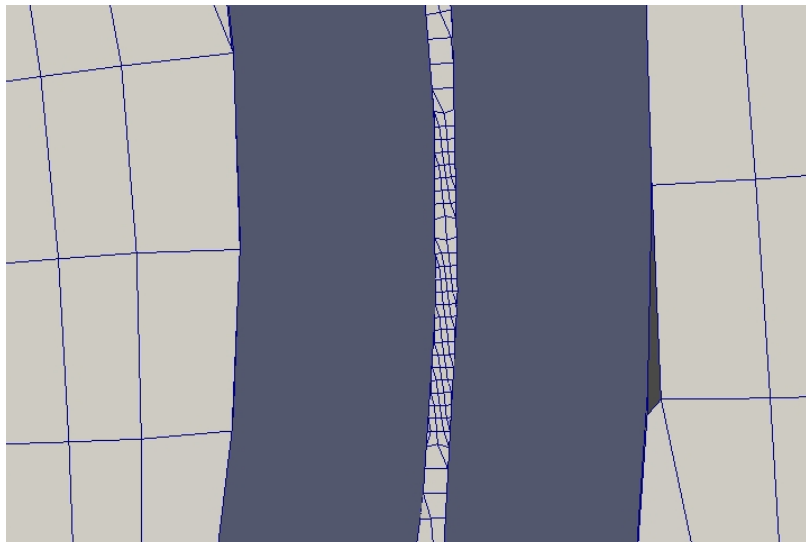


Figura 5.1 – Esempio di *refinement* della *Mesh* tra due superfici vicine

L'analisi è stata effettuata partendo con valori di *gap* molto elevati ed avvicinandosi sempre più alle distanze reali, determinando ogni volta il numero di celle necessarie ad ottenere la *Mesh* ed il tempo computazionale richiesto.

La prima fase dell'analisi è stata condotta considerando un *gap* assiale nullo al fine di considerare soltanto l'influenza della distanza radiale.

Gap radiale [mm]	Gap assiale [mm]	Numero di celle nella <i>Mesh</i>	Tempo di generazione [s]
0,1	0	≈ 338 000	150
0,08	0	≈ 349 000	160
0,05	0	≈ 390 000	190
0,03	0	≈ 435 000	220

Tabella 5.1 – Numero di celle e tempo di generazione della *Mesh* al variare del *gap* radiale

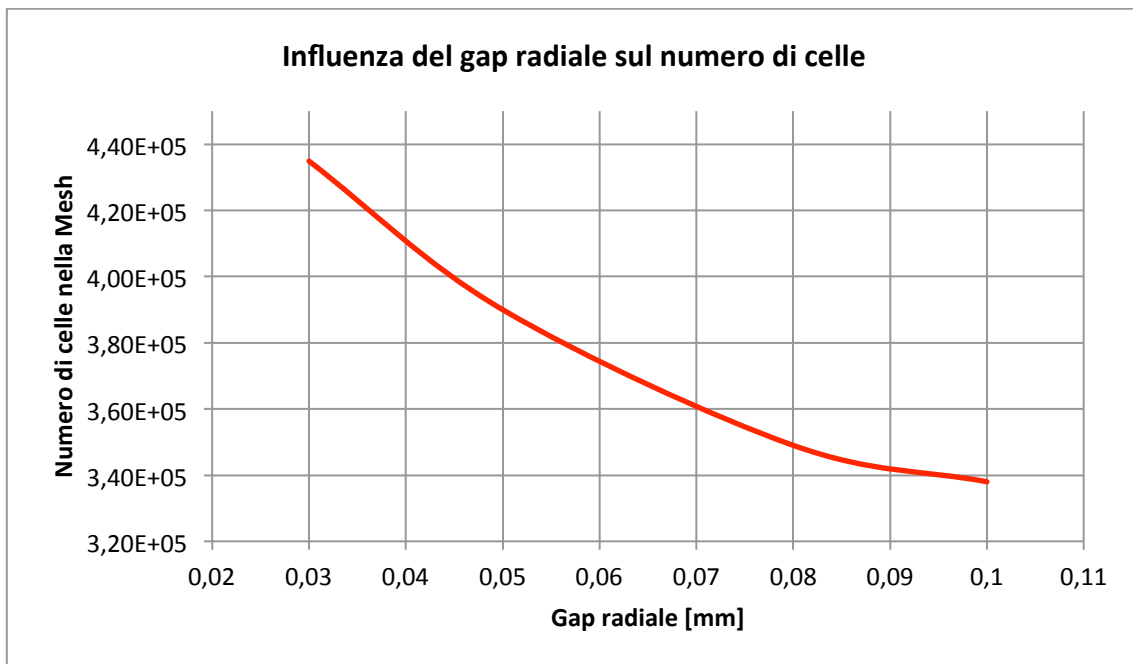


Grafico 5.1 – Influenza del *gap* radiale sul numero di celle nella *Mesh*

Successivamente, considerando il valore minimo di *gap* radiale, che coincide con il valore presente sulla macchina reale, è stata effettuata l'analisi del *gap* assiale.

Gap radiale [mm]	Gap assiale [mm]	Numero di celle nella <i>Mesh</i>	Tempo di generazione [s]
0,03	2	≈ 705 000	350
0,03	1	≈ 1 600 000	1 000
0,03	0,5	≈ 3 800 000	2 600
0,03	0,1	≈ 7 200 000	4 700
0,03	0,05	> 10 000 000	> 7 000

Tabella 5.2 – Numero di celle e tempo di generazione della *Mesh* al variare del *gap* assiale

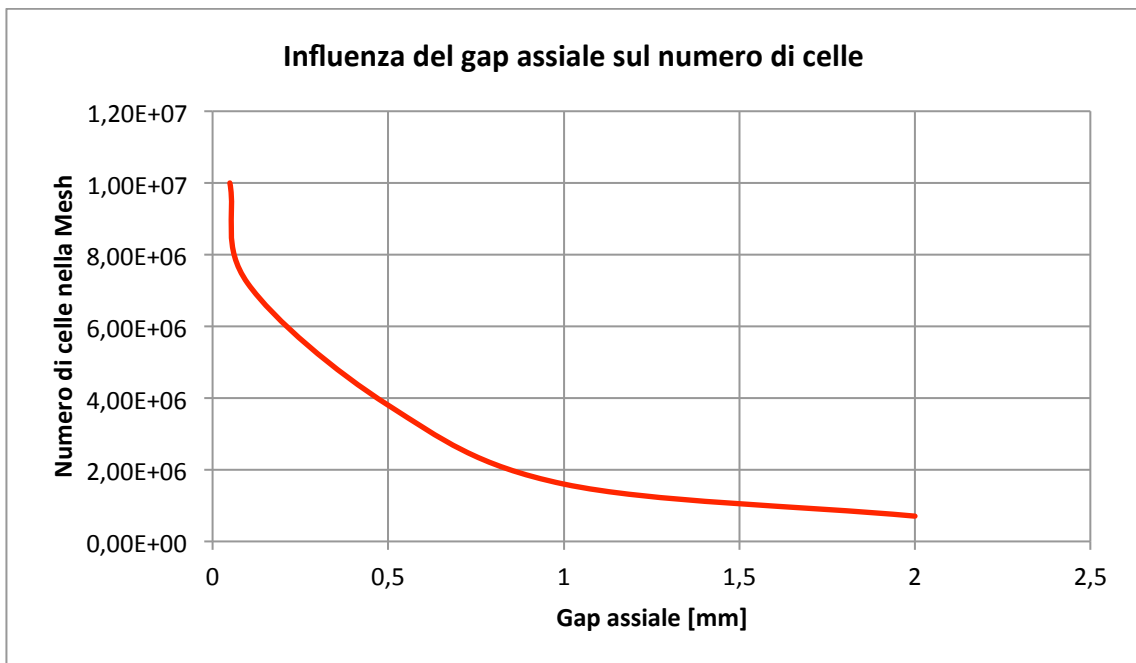


Grafico 5.2 – Influenza del *gap* assiale sul numero di celle nella *Mesh*

L'analisi è stata effettuata utilizzando un singolo processore e, ovviamente, il tempo computazionale richiesto sarà tanto minore quanto maggiore è la potenza del calcolatore e quanto maggiore è il numero di processori impiegati nel processo di generazione della *Mesh*.

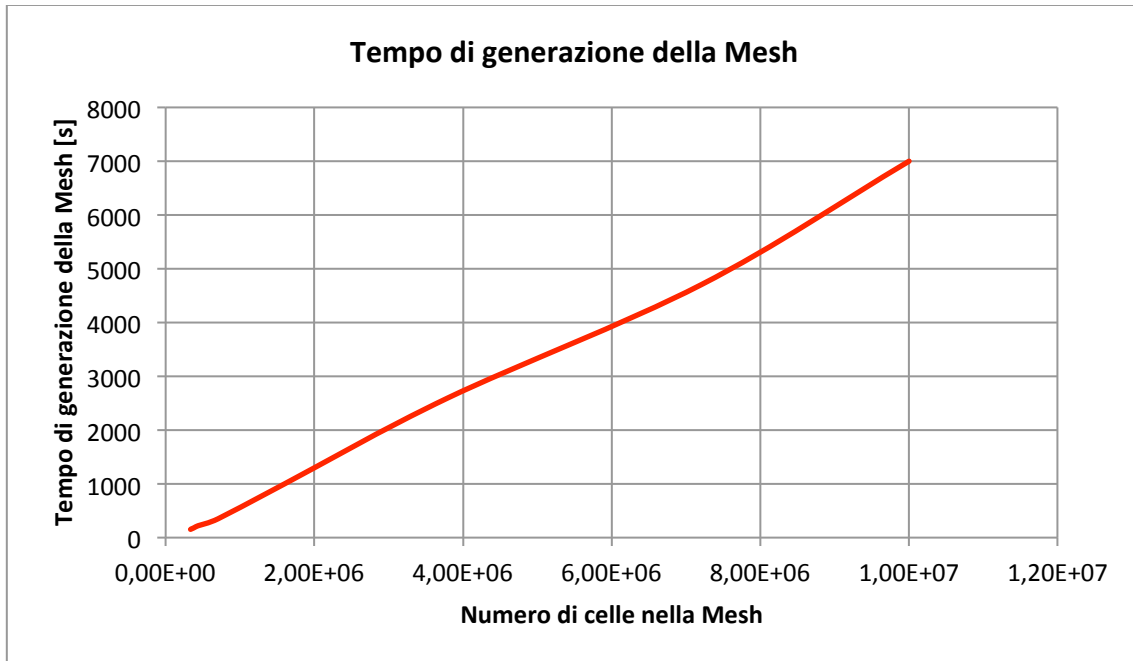


Grafico 5.3 – Tempo di generazione della *Mesh* al variare del numero di celle

CAPITOLO 6

RISULTATI DELLE PROVE

Una volta generate tutte le *Mesh* necessarie a coprire tutti i 360° di un giro completo, viene eseguito il risolutore *rotaryMachineFoam* che permette di determinare l'andamento di pressione, temperatura e velocità all'interno della macchina considerando le caratteristiche fisiche e termodinamiche del fluido reale.

6.1 – CONDIZIONI AL CONTORNO

Prima di eseguire il risolutore vengono definite delle condizioni al contorno che descrivono il valore della temperatura T , della pressione p e della velocità U in opportune sezioni all'istante iniziale del processo di simulazione.

Sezione di *Inlet* del fluido

Nella sezione di ingresso del fluido nella macchina si ipotizzano condizioni ambiente per la temperatura e la pressione, e si impone un gradiente di velocità nullo

$$T_{IN} = 298 \text{ [K]} \quad (6.1)$$

$$p_{IN} = 101325 \text{ [Pa]} \quad (6.2)$$

$$\frac{\partial U_{IN}}{\partial n} = 0 \quad (6.3)$$

Sezione di *Outlet* del fluido

Nella sezione di uscita del fluido dalla macchina si scelgono condizioni al contorno più complesse; si impone un gradiente nullo per la temperatura, si fissa un valore di pressione totale e si utilizza la condizione *pressureInletOutletVelocity* per la velocità che impone automaticamente che il gradiente sia nullo per tutte le componenti tranne quella responsabile dello smaltimento della portata

$$\frac{\partial T_{OUT}}{\partial n} = 0 \quad (6.4)$$

$$p_{OUT} = p_{statica} + \frac{1}{2} \rho |U|^2 = 101325 \text{ [Pa]} \quad (6.5)$$

$$U_{OUT} = \textit{pressureInletOutletVelocity} \quad (6.6)$$

In tutti gli altri punti del dominio di calcolo le variazioni delle grandezze di interesse vengono determinate in automatico dal risolutore durante l'esecuzione della simulazione.

6.2 – CARATTERISTICHE DEL FLUIDO

Per eseguire la simulazione del funzionamento si sceglie di usare aria come fluido di lavoro; per definirne correttamente il comportamento reale il suo calore specifico non può essere considerato costante ma viene determinato in funzione della temperatura attraverso la seguente relazione polinomiale

$$C_p = \bar{R} [a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4] \quad \left[\frac{kJ}{kg K} \right] \quad (6.2)$$

Il valore della costante \bar{R} che compare nell'espressione (6.2) è detta *costante specifica dei gas* ed il suo valore viene determinato come il rapporto tra la *costante universale dei gas* R e la massa molare MM del fluido considerato. Per l'aria questa è pari a

$$\bar{R} = \frac{R}{MM} = \frac{8,314472 \left[\frac{J}{mol K} \right]}{28,96 \left[\frac{g}{mol} \right]} = 287,1 \left[\frac{kJ}{kg K} \right] \quad (6.3)$$

I cinque coefficienti a_0 , a_1 , a_2 , a_3 e a_4 sono tabulati nel database NIST (National Institute of Standards and Technology) in funzione del tipo di fluido e permettono di determinare il calore specifico per temperature comprese tra i 200 °K ed i 6000 °K.

Per le applicazioni in cui si utilizza aria, in genere si considerano solamente temperature inferiori ai 2000 °K.

Coefficienti NIST per aria con temperatura 200 K < T < 1000 K				
a_0	a_1	a_2	a_3	a_4
3,1297	$1,2118 \cdot 10^{-3}$	$-4,0476 \cdot 10^{-7}$	$6,4096 \cdot 10^{-11}$	$-3,7805 \cdot 10^{-15}$
Coefficienti NIST per aria con temperatura 1000 K < T < 2000 K				
a_0	a_1	a_2	a_3	a_4
3,5939	$-8,4193 \cdot 10^{-4}$	$2,0846 \cdot 10^{-6}$	$-5,9384 \cdot 10^{-10}$	$-2,4568 \cdot 10^{-13}$

Tabella 6.1 – Coefficienti NIST per il calcolo del calore specifico dell'aria

6.3 – RISULTATI

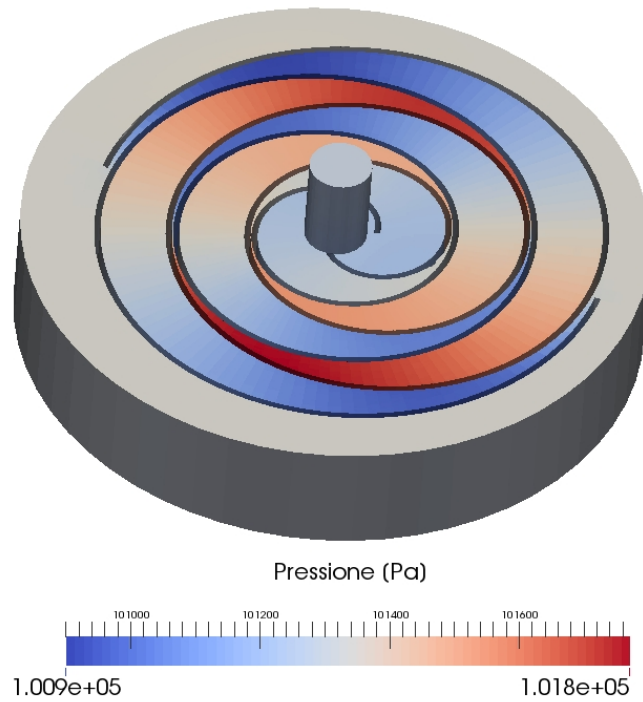


Figura 6.1 – Campo di pressione all’inizio della simulazione (1°)

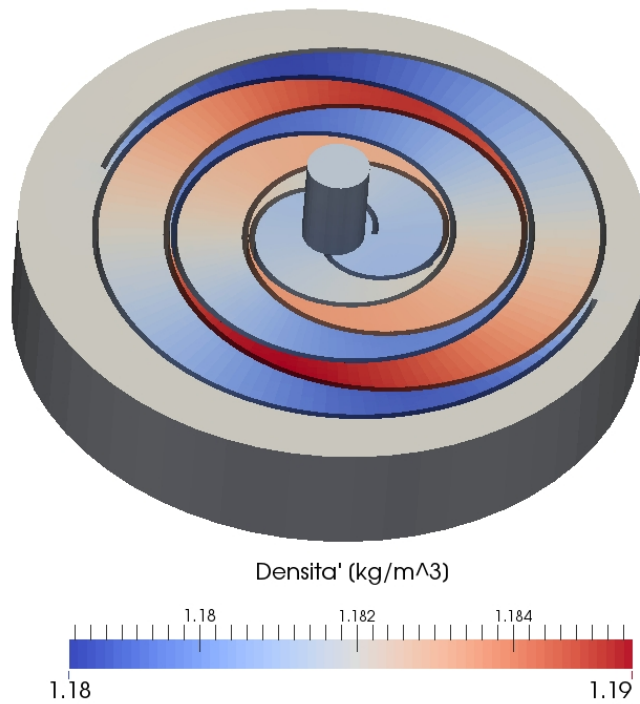


Figura 6.2 – Campo di densità all’inizio della simulazione (1°)

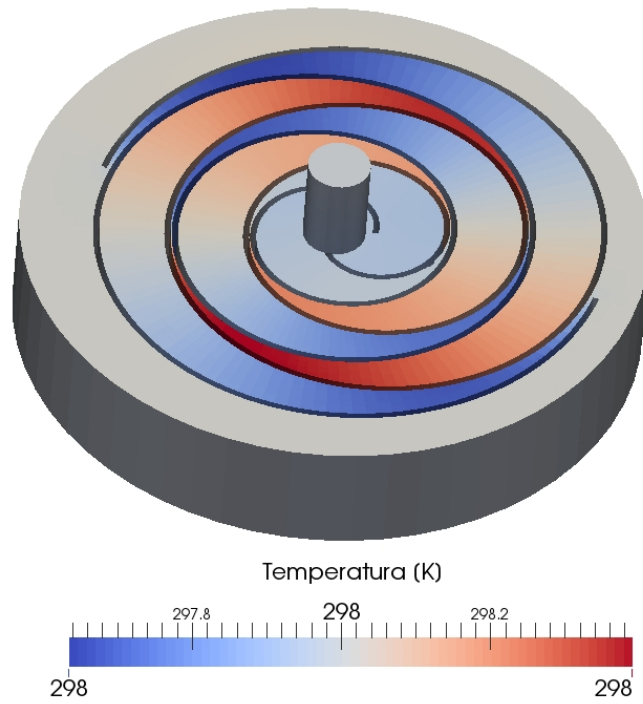


Figura 6.3 – Campo di temperatura all’inizio della simulazione (1°)

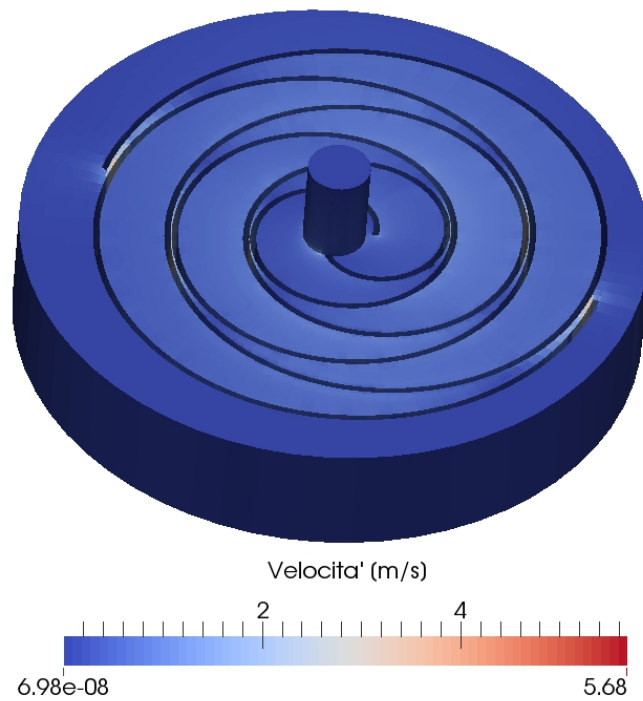


Figura 6.4 – Campo di velocità all’inizio della simulazione (1°)

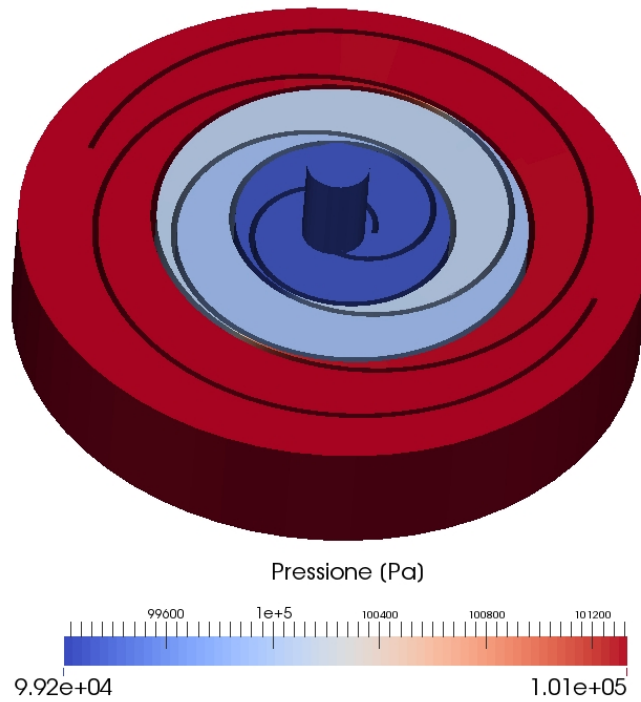


Figura 6.5 – Campo di pressione a 50° di rotazione

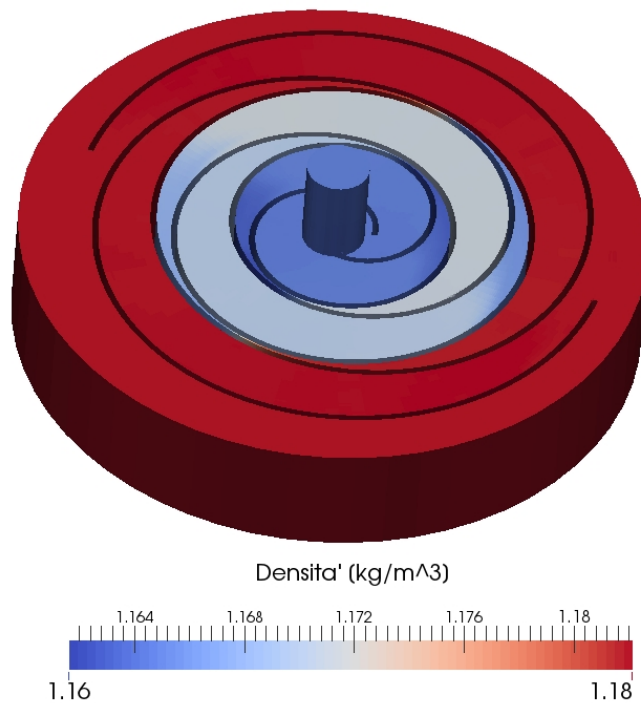


Figura 6.6 – Campo di densità a 50° di rotazione

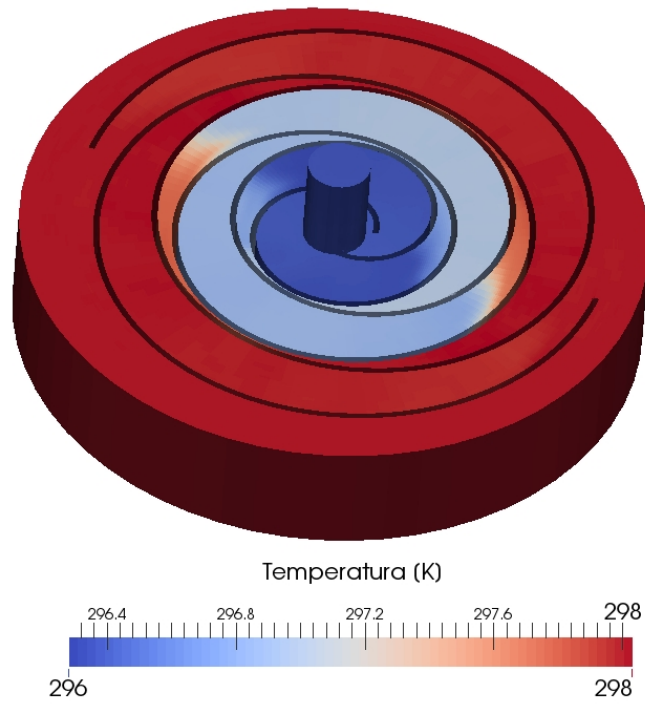


Figura 6.7 – Campo di temperatura a 50° di rotazione

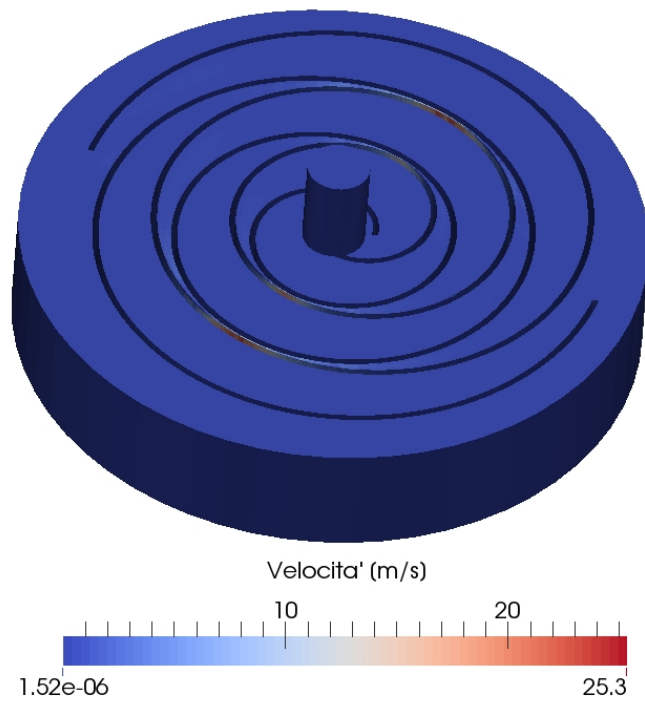


Figura 6.8 – Campo di velocità a 50° di rotazione

La velocità del fluido rimane sempre molto bassa in tutte le zone della macchina tranne nel *gap* tra i due profili dove si verifica il trafileamento del fluido.

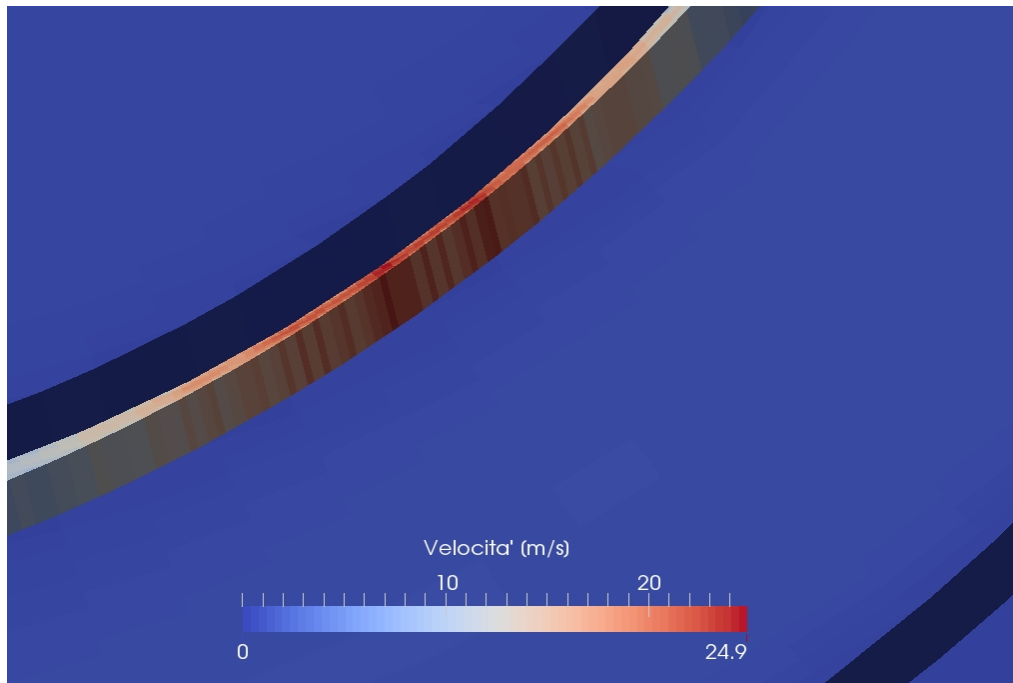


Figura 6.9 – Particolare della velocità nella zona di trafileamento

A causa di problemi numerici legati all'esecuzione del codice, durante la generazione delle *Mesh* vengono create delle celle distorte quando il movimento supera i 50° di rotazione.

Questo peggioramento della qualità della griglia di calcolo influenza negativamente il risolutore che non è in grado di risolvere correttamente le equazioni fluidodinamiche; nelle celle distorte, infatti, si ottengono valori di temperatura molto bassi, nell'ordine dei 4 – 6 K e velocità superiori ai 600 $\left[\frac{m}{s}\right]$.

Valori di questo tipo sono molto lontani dal reale comportamento fisico del fluido e provocano la divergenza della soluzione.

I risultati ottenuti sono però poco significativi perché le condizioni al contorno, soprattutto allo scarico, sono state imposte senza avere a disposizione dati sperimentali sul loro reale valore e, inoltre, la prima orbita della spirale ha soltanto il compito di adeguare tutti i campi di moto al movimento della macchina ed il reale processo di compressione inizia dalla seconda orbita in poi.

Queste considerazioni sono confermate anche dall'andamento della pressione e della temperatura allo scarico che, anziché aumentare, si riducono.

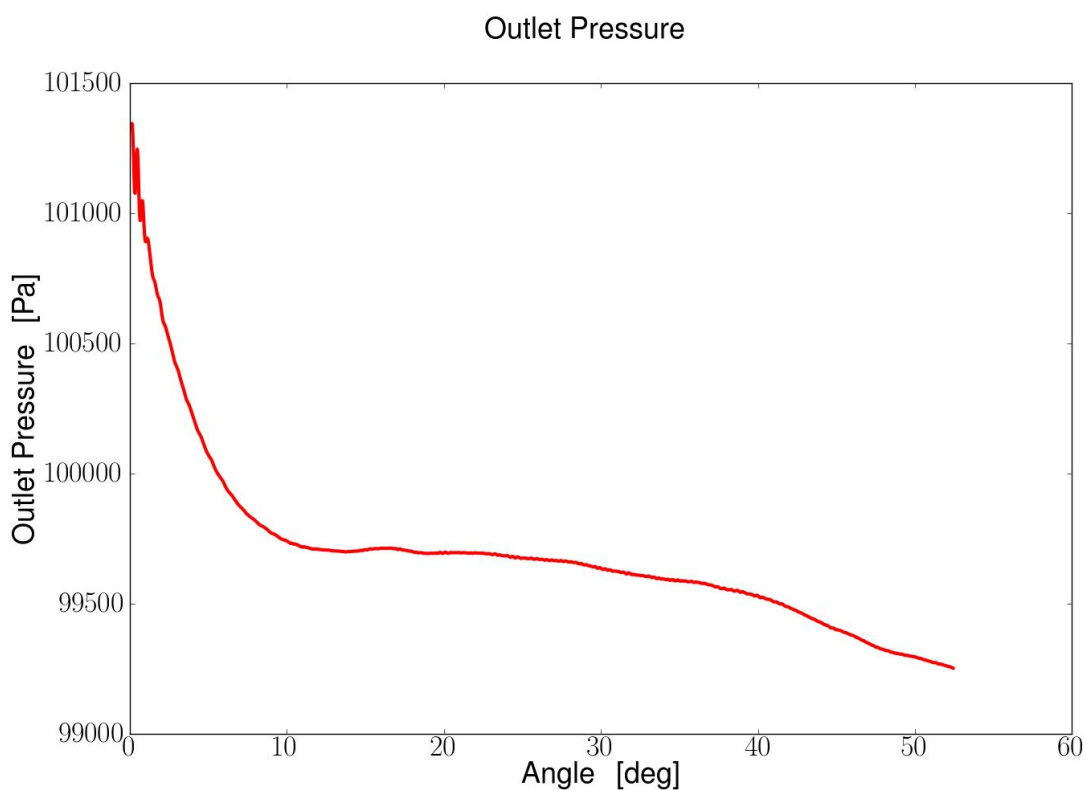


Grafico 6.1 – Andamento della pressione allo scarico

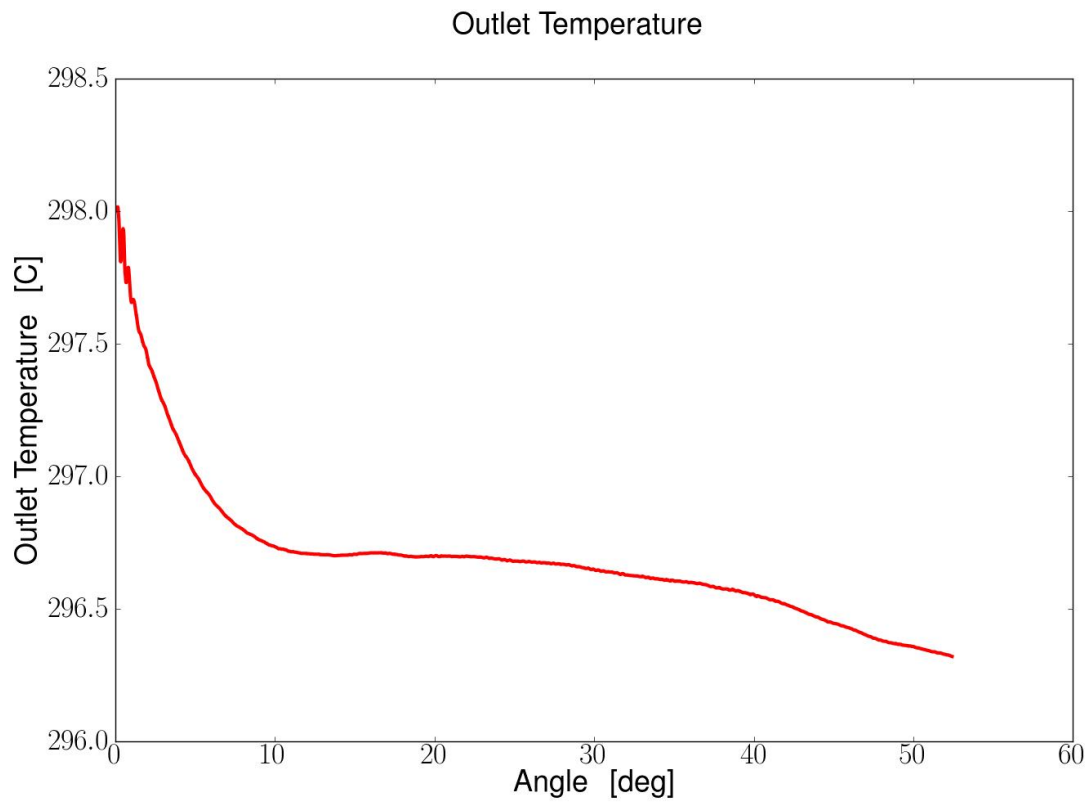


Grafico 6.2 – Andamento della temperatura allo scarico

CAPITOLO 7

CASO 2D

Dopo aver analizzato la macchina reale si prende in esame un caso più semplice con una geometria solamente bidimensionale in cui i profili di statore e rotore vengono creati in modo molto semplice, considerando solamente le due spirali senza ulteriori superfici.

Per eseguire una simulazione 2D e non più 3D, la griglia di calcolo viene impostata per avere soltanto una cella in altezza; in questo modo durante l'esecuzione del risolutore le equazioni di conservazione vengono risolte soltanto nelle due direzioni x e y e non in direzione z .

Con questa simulazione semplificata non si vuole riprodurre il funzionamento reale della macchina ma si vuole soltanto impostare il risolutore in modo che sia possibile utilizzare dei particolari modelli termodinamici che permettono di determinare con maggiore precisione le caratteristiche del fluido reale.

Basandosi su algoritmi sviluppati in precedenti lavori di tesi e già implementati nella libreria *LibRotaryMachine*, il risolutore *rotaryMachineFoam* viene configurato per poter utilizzare i modelli di *Peng-Robinson* e di *Helmholtz* per determinare, rispettivamente, la densità e l'entalpia del fluido.

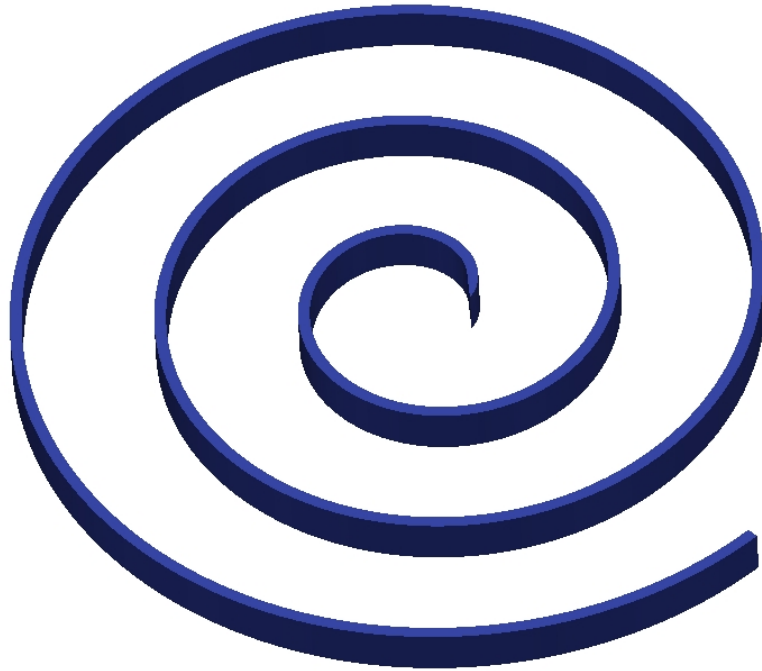


Figura 7.1 – Vista del profilo di statore e rotore usato per il caso 2D

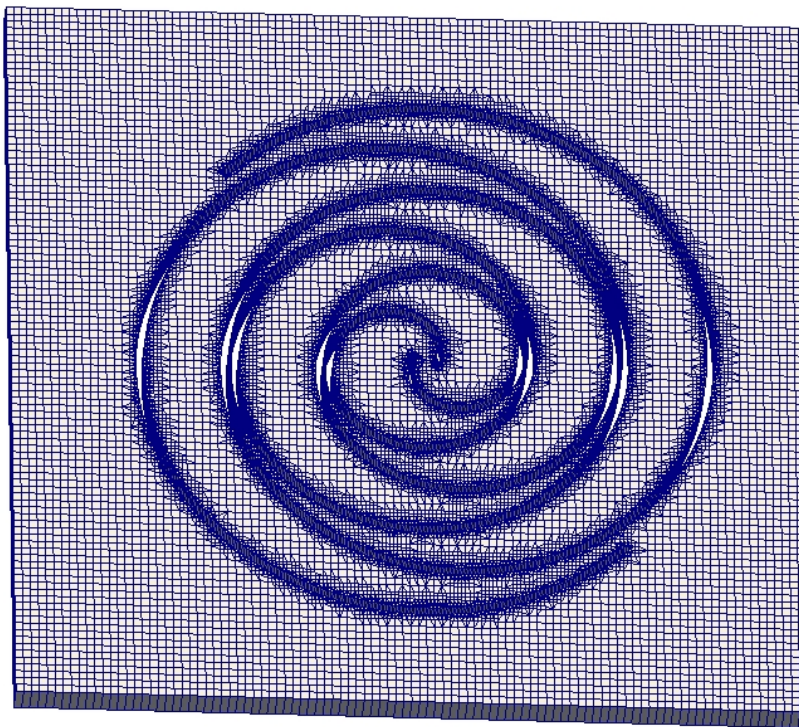


Figura 7.2 – Vista della *Mesh* semplificata usata nel caso 2D

7.1 – CALCOLO DELLA DENSITÀ

Per determinare la densità di un fluido il cui comportamento non può essere assimilato a quello di un gas perfetto, il metodo più diffuso prevede l'impiego dell'*equazione di Peng-Robinson*, ovvero una particolare formulazione della classica equazione di stato che permette di avere una stima più precisa della densità soprattutto quando le condizioni sono vicine a quelle critiche.

$$p = \frac{R T}{v - b} - \frac{\alpha(T) a}{v^2 + 2 b v - b^2} \quad (7.1)$$

in cui:

- v è il volume specifico del fluido
- $a = \frac{0,45724 R^2 T_c^2}{p_c}$
- $b = \frac{0,0778 R T_c}{p_c}$
- T_c è la temperatura critica del fluido
- p_c è la pressione critica del fluido
- $\alpha = 0,37464 + 1,54226 \omega - 0,26992 \omega^2$

Il parametro ω viene definito *fattore acentrico di Pitzer* e rappresenta una particolare formulazione del fattore di comprimibilità. Il suo valore viene determinato come

$$\omega = -\log_{10}(p_r^{sat})_{T_r=0,7} \quad (7.2)$$

dove:

- $p_r^{sat} = \frac{p^{sat}}{p_c}$ è la pressione di saturazione del vapore ridotta
- $T_r = \frac{T}{T_c}$ è la temperatura ridotta

7.2 – CALCOLO DELL'ENTALPIA

Quando il comportamento del fluido può essere assimilato a quello di un gas perfetto, ovvero quando il suo calore specifico C_p non dipende dalla temperatura, l'entalpia può essere determinata semplicemente usando la relazione

$$h = C_p T \quad (7.3)$$

Per considerare l'effetto reale del fluido bisogna ricorrere a relazioni più complesse come, ad esempio, l'*equazione di stato esplicita nell'energia di Helmholtz* che fornisce risultati molto accurati ed è allo stesso tempo semplice da usare.

$$a(\rho, T) = a^0(\rho, T) + a^r(\rho, T) \quad (7.4)$$

Da questa relazione si vede come l'energia di *Helmholtz* $a(\rho, T)$ sia funzione di due parametri: $a^0(\rho, T)$ che rappresenta il contributo fornito del gas ideale, a cui si aggiunge una parte $a^r(\rho, T)$ dovuta al comportamento reale del gas.

Utilizzando questa relazione si ha il vantaggio di poter scrivere tutte le proprietà termodinamiche del fluido semplicemente tramite le derivate dell'energia $a(\rho, T)$, ad esempio la pressione può essere calcolata come

$$p = \rho^2 \left(\frac{\partial a}{\partial \rho} \right)_T \quad (7.5)$$

Nelle applicazioni pratiche si preferisce utilizzare la relazione dell'*energia di Helmholtz* in forma adimensionale, ovvero rapportando i valori di densità e temperatura ai corrispondenti valori critici.

$$\frac{a(\rho, T)}{R T} = \alpha(\delta, \tau) = \alpha^0(\delta, \tau) + \alpha^r(\delta, \tau) \quad (7.6)$$

dove:

- $\delta = \frac{\rho}{\rho_c}$ è la densità adimensionale
- $\tau = \frac{T_c}{T}$ è la temperatura adimensionale

Per quanto riguarda la parte ideale dell'energia $a^0(\rho, T)$, questa può essere determinata tramite la relazione

$$a^0 = h^0 - R T - T s^0 \quad (7.6)$$

in cui l'entalpia ideale h^0 e l'entropia ideale s^0 sono pari rispettivamente a

$$h^0 = h_0^0 + \int_{T_0}^T C_P^0 dT \quad (7.7)$$

$$s^0 = s_0^0 + \int_{T_0}^T \frac{C_P^0}{T} dT - R \ln\left(\frac{\rho T}{\rho_0 T_0}\right) \quad (7.8)$$

in cui:

- C_P^0 è il calore specifico del gas ideale
- ρ_0 è la densità determinata in corrispondenza della temperatura T_0
- h_0^0 e s_0^0 vengono scelti in base a precise convenzioni che dipendono dal tipo di fluido considerato

Usando queste relazioni, la parte ideale dell'energia di Helmholtz può essere riscritta come

$$a^0 = h_0^0 + \int_{T_0}^T C_P^0 dT - R T - T \left[s_0^0 + \int_{T_0}^T \frac{C_P^0}{T} dT - R \ln\left(\frac{\rho T}{\rho_0 T_0}\right) \right] \quad (7.9)$$

e in forma adimensionale

$$\alpha^0 = \frac{h_0^0 \tau}{R T_C} - \frac{s_0^0}{R} - 1 + \ln\left(\frac{\delta \tau_0}{\delta_0 \tau}\right) - \frac{\tau}{R} \int_{\tau_0}^{\tau} \frac{C_P^0}{\tau^2} d\tau + \frac{1}{R} \int_{\tau_0}^{\tau} \frac{C_P^0}{\tau} d\tau \quad (7.10)$$

Il calore specifico ideale C_P^0 viene determinato usando una relazione che, tipicamente, è ottenuta interpolando statisticamente i valori assunti dal calore specifico in diverse condizioni di pressione e di temperatura.

$$C_P^0 = R \left[c_0 + c_1 \left(\frac{T}{K}\right)^{c_2} + \sum_{k=1}^5 v_k \left(\frac{u_k}{T}\right)^2 \frac{e^{\frac{u_k}{T}}}{\left(e^{\frac{u_k}{T}} - 1\right)^2} \right] \quad (7.11)$$

Usando questa espressione per il calore specifico la parte ideale dell'energia di *Helmholtz* può essere scritta come

$$\alpha^0 = a_1 + a_2 \tau + \ln(\delta) + (c_0 - 1) \ln(\tau) + \frac{c_1 \left(\frac{T_C}{K}\right)^{c_2}}{c_2 (c_2 + 1)} \tau^{-c_2} + \sum_{k=1}^5 v_k \left(1 - e^{-\frac{u_k \tau}{T_C}}\right) \quad (7.12)$$

I coefficienti c_0 , c_1 , c_2 , v_k , u_k , a_1 e a_2 che compaiono nelle equazioni (7.11) e (7.12) sono tabulati in funzione del tipo di fluido.

	v_1	$\frac{u_1}{K}$	v_2	$\frac{u_2}{K}$	v_3	$\frac{u_3}{K}$	v_4	$\frac{u_4}{K}$
Acetone	3,7072	310	7,0675	3480	11,012	1576	\	\
Toluene	1,6994	190	8,0577	797	17,059	1619	8,4567	3072
R-141b	6,8978	502	7,8157	1571	3,0239	4603	\	\
R-245fa	5,5728	222	10,385	1010	12,554	2450	\	\

Tabella 7.1 – Esempio di coefficienti per il calcolo dell'energia di *Helmholtz* ideale

	c_0	c_1	c_2	a_1	a_2
Acetone	4	\	\	-9,488365997	7,1422719708
Toluene	4	\	\	-3,5241174832	1,1360823464
R-141b	4	\	\	-15,5074814985	9,1871858933
R-245fa	4	\	\	-13,4283638514	9,78236538

Tabella 7.2 – Esempio di coefficienti per il calcolo dell'energia di *Helmholtz* ideale

Per quanto riguarda invece il comportamento reale del gas, il suo contributo $a^r(\rho, T)$ viene determinato usando dei modelli empirici i cui coefficienti vengono ottenuti attraverso prove sperimentali. In forma adimensionale, la parte reale dell'energia di *Helmholtz* può essere scritta come

$$\alpha^r = \sum n_k \delta^{i_k} \tau^{j_k} + \sum n_k \delta^{i_k} \tau^{j_k} e^{-\delta^{l_k}} \quad (7.11)$$

dove:

- n_k sono valori tabulati in funzione del tipo di fluido
- i_k, j_k e l_k sono numeri arbitrari generalmente maggiori di zero

L'espressione del contributo reale può essere scritta esplicitando le sommatorie e assume due forme differenti a seconda che il fluido considerato sia un *fluido polare* o un *fluido non polare*.

$$\begin{aligned} \alpha_{polare}^r = & n_1 \delta \tau^{0,25} + n_2 \delta \tau^{1,25} + n_3 \delta \tau^{1,5} + n_4 \delta^3 \tau^{0,25} + \\ & + n_5 \delta^7 \tau^{0,875} + n_6 \delta \tau^{2,375} e^{-\delta} + n_7 \delta^2 \tau^2 e^{-\delta} + n_8 \delta^5 \tau^{2,125} e^{-\delta} + \\ & + n_9 \delta \tau^{3,5} e^{-\delta^2} + n_{10} \delta \tau^{6,5} e^{-\delta^2} + n_{11} \delta^4 \tau^{4,75} e^{-\delta^2} + n_{12} \delta^2 \tau^{12,5} e^{-\delta^3} \end{aligned} \quad (7.12)$$

$$\begin{aligned} \alpha_{non-polare}^r = & n_1 \delta \tau^{0,25} + n_2 \delta \tau^{1,125} + n_3 \delta \tau^{1,5} + n_4 \delta^2 \tau^{1,375} + \\ & + n_5 \delta^3 \tau^{0,25} + n_6 \delta^7 \tau^{0,875} + n_7 \delta^2 \tau^{0,625} e^{-\delta} + n_8 \delta^5 \tau^{1,75} e^{-\delta} + \\ & + n_9 \delta \tau^{3,625} e^{-\delta^2} + n_{10} \delta^4 \tau^{3,625} e^{-\delta^2} + n_{11} \delta^3 \tau^{14,5} e^{-\delta^3} + n_{12} \delta^4 \tau^{12} e^{-\delta^3} \end{aligned} \quad (7.13)$$

	Acetone	R-141b	R-245fa
n_1	0,90041	1,4169	1,2904
n_2	-2,1276	-3,6799	-3,2154
n_3	-0,083409	1,3469	0,50693
n_4	0,065683	0,083329	0,093148
n_5	0,00016527	0,00025137	0,00027638
n_6	-0,039663	0,3272	0,71458
n_7	0,72085	0,46946	0,87252
n_8	0,0092318	-0,029829	-0,015077
n_9	-0,17217	-0,31621	-0,40645
n_{10}	-0,14961	-0,026219	-0,11701
n_{11}	-0,076124	-0,078043	-0,13602
n_{12}	-0,018166	-0,020498	-0,022952

Tabella 7.3 – Esempio di coefficienti per il calcolo dell'energia di *Helmholtz* reale

7.3 – VERIFICA DI FUNZIONAMENTO

Per verificare il funzionamento del risolutore *rotaryMachineFoam* così modificato, si impongono condizioni ambiente nella sezione di ingresso della macchina ($T = 298\text{ K}$ e $P = 101325\text{ Pa}$) e si sceglie di usare come fluido di lavoro il freon R-245fa che ha le seguenti caratteristiche:

- Massa molare: $w = 134,05 \left[\frac{\text{g}}{\text{mol}} \right]$
- Calore specifico: $C_p = 857 \left[\frac{\text{J}}{\text{kg K}} \right]$
- Densità: $\rho = 5,84 \left[\frac{\text{kg}}{\text{m}^3} \right]$
- Densità critica: $\rho_c = 517 \left[\frac{\text{kg}}{\text{m}^3} \right]$
- Temperatura critica: $T_c = 427,2 \text{ [K]}$
- Pressione critica: $P_c = 3,64 \cdot 10^6 \text{ [Pa]}$

Come esempio dei risultati ottenuti si riportano i valori di pressione, temperatura, densità e velocità del fluido determinati all'inizio della simulazione ed in corrispondenza di un angolo di rotazione del rotore di 90° .

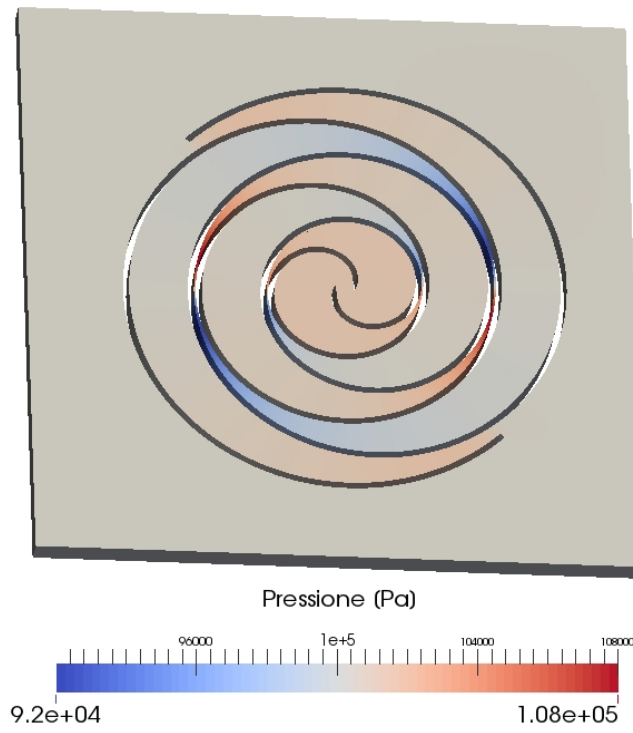


Figura 7.3 – Campo di pressione all'inizio del movimento

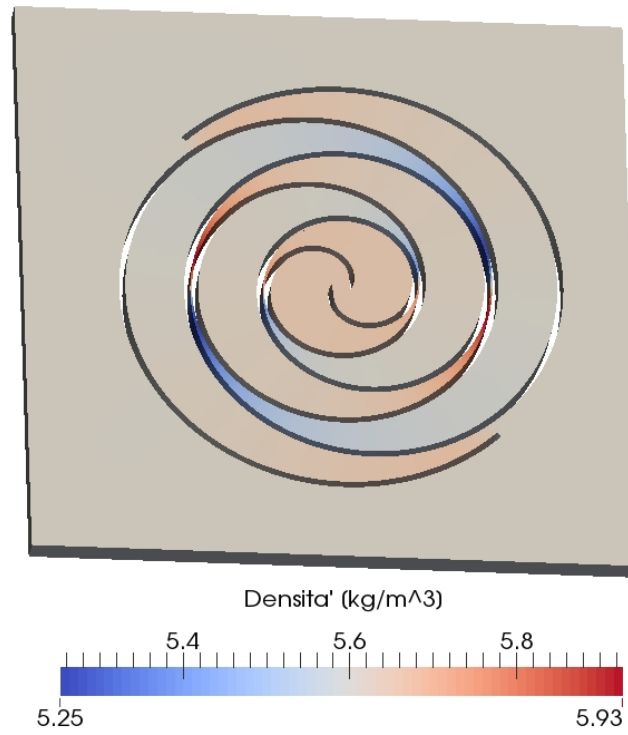


Figura 7.4 – Campo di densità all'inizio del movimento

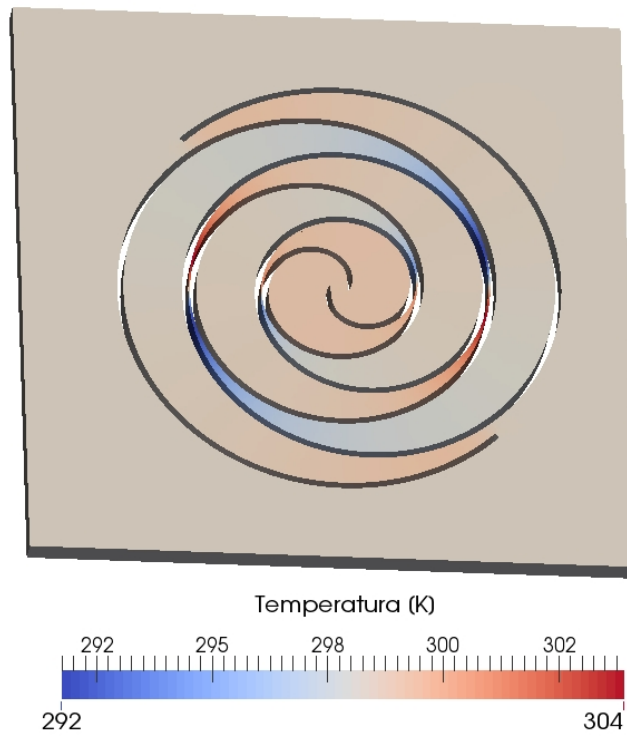


Figura 7.5 – Campo di temperatura all'inizio del movimento

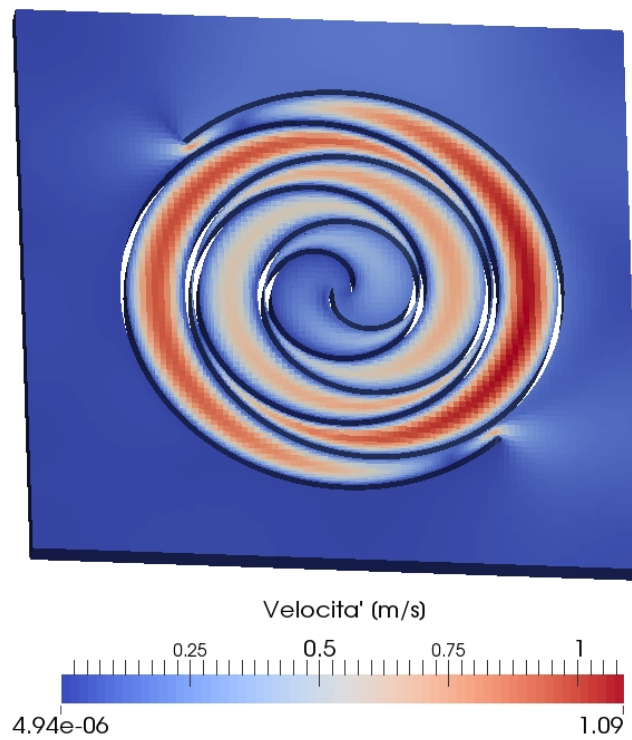


Figura 7.6 – Campo di velocità all'inizio del movimento

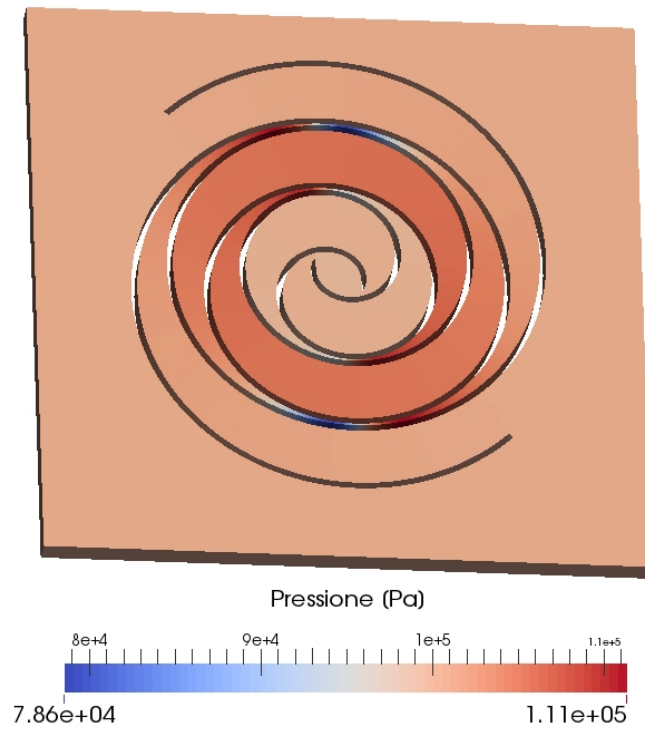


Figura 7.7 – Campo di pressione a 90° di rotazione

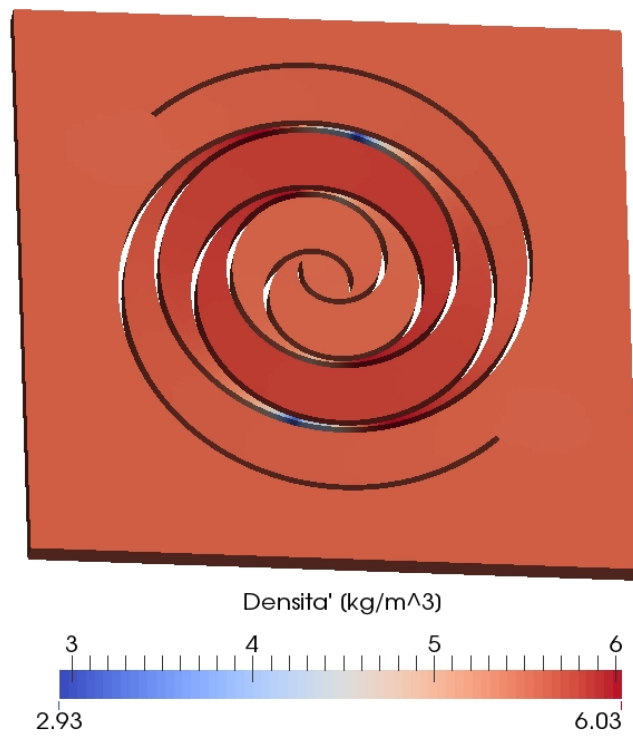


Figura 7.8 – Campo di densità a 90° di rotazione

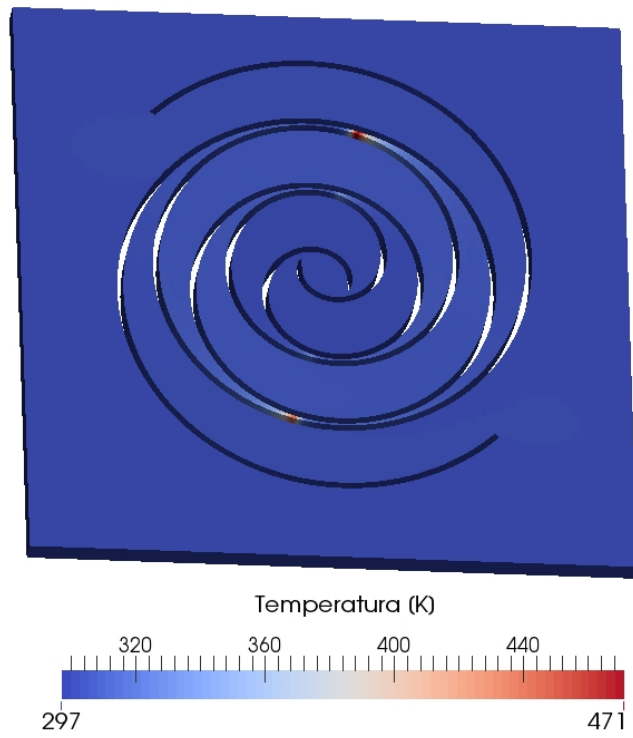


Figura 7.9 – Campo di temperatura a 90° di rotazione

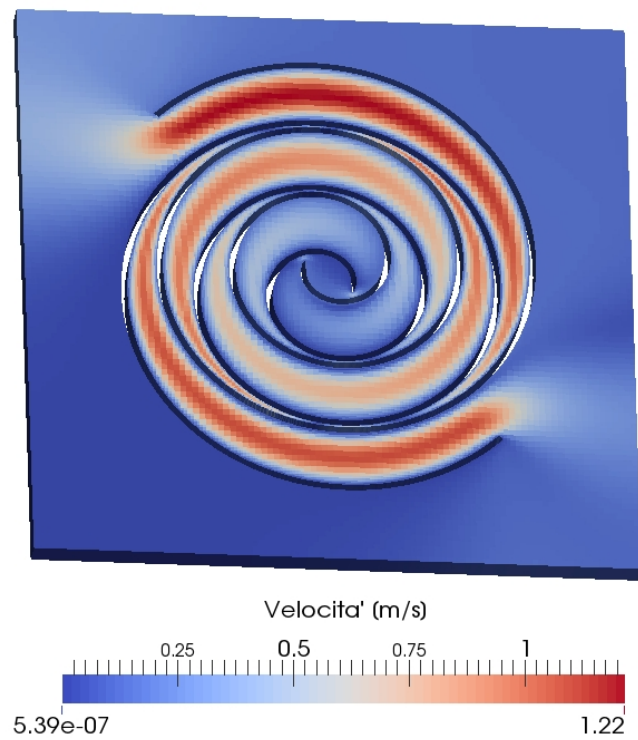


Figura 7.10 – Campo di velocità a 90° di rotazione

SVILUPPI FUTURI

Per completare il lavoro sviluppato in questa tesi si possono effettuare diverse operazioni, in particolare:

- Scegliere ed impostare una nuova strategia di generazione delle *Mesh* che permetta di ottenere una qualità della griglia di calcolo molto elevata e che non peggiori durante il movimento, al fine di assicurare una corretta convergenza della soluzione.
- Estendere il movimento della macchina ed arrivare a coprire tre o più orbite al fine di ottenere risultati più attendibili.
- Impostare delle nuove condizioni al contorno che permettano di riprodurre il più fedelmente possibile le reali condizioni di ingresso ed uscita del fluido, con la possibilità di riferirsi a valori sperimentali.
- Effettuare simulazioni con diversi tipi di fluido.

ALLEGATI

ALLEGATO 1

Dizionario *blockMeshDict* per la creazione del dominio di calcolo

```

/*-----* - C++ - *-----*/
|=====|
| \ \ \ \ / | Field | OpenFOAM: The Open Source CFD Toolbox
| \ \ \ \ / | Operation | Version: 2.3.0
| \ \ \ \ / | And | Web: www.OpenFOAM.org
| \ \ \ \ / | Manipulation |
/*-----*-----*/

```

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict
}

```

```

// *****

```

```

convertToMeters 0.001;

```

```

vertices
(
    //Bottom surface
    (-10 -10 18) //Point 0
    (10 -10 18) //Point 1
    (10 10 18) //Point 2
    (-10 10 18) //Point 3
    (-70.711 -70.711 18) //Point 4
    (70.711 -70.711 18) //Point 5
    (70.711 70.711 18) //Point 6
    (-70.711 70.711 18) //Point 7

    //Top surface
    (-10 -10 19) //Point 8
    (10 -10 19) //Point 9
    (10 10 19) //Point 10
    (-10 10 19) //Point 11
    (-70.711 -70.711 19) //Point 12
    (70.711 -70.711 19) //Point 13
    (70.711 70.711 19) //Point 14
    (-70.711 70.711 19) //Point 15
)

```

```

//Discharge duct
(-10 -10 49) //Point 16
(10 -10 49) //Point 17
(10 10 49) //Point 18
(-10 10 49) //Point 19
);

blocks
(
  hex (4 5 1 0 12 13 9 8) (15 70 1) simpleGrading (1 1 1) //Block 1
  hex (1 5 6 2 9 13 14 10) (70 15 1) simpleGrading (1 1 1) //Block 2
  hex (3 2 6 7 11 10 14 15) (15 70 1) simpleGrading (1 1 1) //Block 3
  hex (4 0 3 7 12 8 11 15) (70 15 1) simpleGrading (1 1 1) //Block 4
  hex (0 1 2 3 8 9 10 11) (15 15 1) simpleGrading (1 1 1) //Block 5
  hex (8 9 10 11 16 17 18 19) (15 15 10) simpleGrading (1 1 1) //Discharge
);

edges
(
  arc 4 5 (0 -100 18)
  arc 5 6 (100 0 18)
  arc 6 7 (0 100 18)
  arc 7 4 (-100 0 18)
  arc 12 13 (0 -100 19)
  arc 13 14 (100 0 19)
  arc 14 15 (0 100 19)
  arc 15 12 (-100 0 19)
);

boundary
(
  bottom
  {
    type wall;
    faces
    (
      (4 5 1 0)
      (1 5 6 2)
      (3 2 6 7)
      (4 0 3 7)
      (0 1 2 3)
    );
  }
);

```

```
top
{
    type wall;
    faces
    (
        (12 13 9 8)
        (9 13 14 10)
        (11 10 14 15)
        (12 8 11 15)
    );
}

inlet
{
    type patch;
    faces
    (
        (4 5 13 12)
        (5 6 14 13)
        (6 7 15 14)
        (7 4 12 15)
    );
}

outlet
{
    type patch;
    faces
    (
        (16 17 18 19)
    );
}

);

mergePatchPairs
(
);
```


ALLEGATO 2

Dizionario *snappyHexMeshDict* per il processo di *snap* della griglia di calcolo

```

/*-----* - C++ - *-----*/
|=====|
| \ \ \ \ / | Field | OpenFOAM: The Open Source CFD Toolbox |
| \ \ \ \ / | Operation | Version: 2.3.0 |
| \ \ \ \ / | And | Web: www.OpenFOAM.org |
| \ \ \ \ / | Manipulation |
|-----*-----*/

```

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       snappyHexMeshDict
}

```

```

// *****//

```

```

// Which of the steps to run
castellatedMesh    true;
snap               true;
addLayers          false;

```

```

//Geometry

```

```

//Definition of all surfaces. All surfaces are of class searchableSurface. Surfaces are
//used:

```

```

//- to specify refinement for any mesh cell intersecting it
//- to specify refinement for any mesh cell inside/outside/near
//- to 'snap' the mesh boundary to the surface

```

```

geometry
{
    stator.stl
    {
        type triSurfaceMesh;
        name stator;
    }
}

```

```

regions
{
    patch2
    {
        name duct;
    }
    patch4
    {
        name top;
    }
    patch5
    {
        name top;
    }
    patch6
    {
        name top;
    }
    patch7
    {
        name stator;
    }
    patch8
    {
        name stator;
    }
    patch9
    {
        name stator;
    }
    patch10
    {
        name stator;
    }
    patch11
    {
        name stator;
    }
}

rotor.stl
{
    type triSurfaceMesh;
    name rotor;
}
}

```

```
//Settings for the castellatedMesh generation
castellatedMeshControls
{
    //Refinement parameters
    //~~~~~

    //If local number of cells is >= maxLocalCells on any processor switches
    //from refinement followed by balancing (current method) to (weighted)
    //balancing before refinement
    maxLocalCells 100000;

    //Overall cell limit (approximately). Refinement will stop immediately
    //upon reaching this number so a refinement level might not complete.
    //Note that this is the number of cells before removing the part which is not
    //'visible' from the keepPoint. The final number of cells might actually be a
    //lot less
    maxGlobalCells 10000000;

    //The surface refinement loop might spend lots of iterations refining just a
    //few cells. This setting will cause refinement to stop if <= minimumRefine
    //are selected for refinement. Note: it will at least do one iteration (unless
    //the number of cells to refine is 0)
    minRefinementCells 10;

    //Number of buffer layers between different levels. 1 means normal 2:1
    //refinement restriction, larger means slower refinement
    nCellsBetweenLevels 1;

    //Explicit feature edge refinement
    //~~~~~

    //Specifies a level for any cell intersected by its edges. This is a
    //featureEdgeMesh, read from constant/triSurface for now
    features
    (
        {
            file "stator.eMesh";
            level 0;
        }

        {
            file "rotor.eMesh";
            level 0;
        }
    );
}
```

```

//Surface based refinement
//~~~~~

//Specifies two levels for every surface. The first is the minimum level,
//every cell intersecting a surface gets refined up to the minimum level.
//The second level is the maximum level. Cells that 'see' multiple
//intersections where the intersections make an angle >
//resolveFeatureAngle get refined up to the maximum level
refinementSurfaces
{
    stator
    {
        level (1 1);
        gapLevelIncrement 4;
    }

    rotor
    {
        level (1 1);
        gapLevelIncrement 4;
    }
}

//Resolve sharp angles
resolveFeatureAngle 30;

//Region-wise refinement
//~~~~~

//Specifies refinement level for cells in relation to a surface. One of three
//modes:
//- distance. 'levels' specifies per distance to the surface the wanted
//refinement
//- level. The distances need to be specified in descending order
//- inside. 'levels' is only one entry and only the level is used. All cells inside
//the surface get refined up to the level. The surface needs to be closed for
//this to be possible
//- outside. Same but cells outside
refinementRegions
{
}

```



```

// Mesh selection
//~~~~~

//After refinement patches get added for all refinementSurfaces and all
//cells intersecting the surfaces get put into these patches. The section
//reachable from the locationInMesh is kept. NOTE: This point should never
//be on a face, always inside a cell, even after refinement
locationInMesh (0 0.00084115061137429 0.0185);

//Whether any faceZones (as specified in the refinementSurfaces) are only
//on the boundary of corresponding cellZones or also allow free-standing
//zone faces. Not used if there are no faceZones
allowFreeStandingZoneFaces false;
}

//Settings for the snapping
snapControls
{
    //Number of patch smoothing iterations before finding correspondence to
    //surface
    nSmoothPatch 3;

    //Relative distance for points to be attracted by surface feature point or
    //edge. True distance is this factor times local maximum edge length
    tolerance 5.0;

    //Number of mesh displacement relaxation iterations
    nSolveIter 1;

    //Maximum number of snapping relaxation iterations. Should stop before
    //upon reaching a correct mesh
    nRelaxIter 6;

    //Feature snapping

    //Number of feature edge snapping iterations. Leave out altogether to
    //disable
    nFeatureSnapIter 8;

    //Detect (geometric only) features by sampling the surface (default=false)
    implicitFeatureSnap false;

    //Use castellatedMeshControls::features (default = true)
    explicitFeatureSnap true;
}

```

```

//Detect points on multiple surfaces (only for explicitFeatureSnap)
multiRegionFeatureSnap false;
}

//Settings for the layer addition
addLayersControls
{
    relativeSizes true;

    //Per final patch (so not geometry!) the layer information
    layers
    {
    }

    //Expansion factor for layer mesh
    expansionRatio 1.3;

    //Wanted thickness of final added cell layer. If multiple layers is the
    //thickness of the layer furthest away from the wall. Relative to undistorted
    //size of cell outside layer
    finalLayerThickness 1;

    //Minimum thickness of cell layer. If for any reason layer cannot be above
    //minThickness do not add layer. Relative to undistorted size of cell outside
    //layer
    minThickness 0.1;

    //If points get not extruded do nGrow layers of connected faces that are
    //also not grown. This helps convergence of the layer addition process
    //close to features.
    nGrow 0;

    //Advanced settings

    //When not to extrude surface. 0 is flat surface, 90 is when two faces are
    //perpendicular
    featureAngle 30;

    //Maximum number of snapping relaxation iterations. Should stop before
    //upon reaching a correct mesh
    nRelaxIter 3;

    //Number of smoothing iterations of surface normals
    nSmoothSurfaceNormals 1;
}

```

```

//Number of smoothing iterations of interior mesh movement direction
nSmoothNormals 3;

//Smooth layer thickness over surface patches
nSmoothThickness 2;

//Stop layer growth on highly warped cells
maxFaceThicknessRatio 0.5;

//Reduce layer growth where ratio thickness to medial distance is large
maxThicknessToMedialRatio 1;

//Angle used to pick up medial axis points
minMedianAxisAngle 90;

//Create buffer region for new layer terminations
nBufferCellsNoExtrude 0;

//Overall max number of layer addition iterations. The mesher will exit if it
//reaches this number of iterations; possibly with an illegal mesh
nLayerIter 50;
}

//Generic mesh quality settings. At any undoable phase these determine where to
//undo
meshQualityControls
{
    //Maximum non-orthogonality allowed. Set to 180 to disable
    maxNonOrtho 90;

    //Max skewness allowed. Set to <0 to disable
    maxBoundarySkewness 20;
    maxInternalSkewness 4;

    //Max concaveness allowed. Is angle (in degrees) below which concavity is
    //allowed. 0 is straight face, <0 would be convex face. Set to 180 to disable
    maxConcave 70;

    //Minimum pyramid volume. Is absolute volume of cell pyramid. Set to very
    //negative number (e.g. -1E30) to disable
    minVol 0;

    //Minimum quality of the tet formed by the face-centre and variable base
    //point minimum decomposition triangles and the cell centre. Set to very
    //negative number (e.g. -1E30) to disable
    minTetQuality -1e50;
}

```

```
//Minimum face area. Set to <0 to disable
minArea -1;

//Minimum face twist. Set to <-1 to disable. dot product of face normal and
//face centre triangles normal
minTwist 0.01;

//Minimum normalised cell determinant
minDeterminant 0.001;

//MinFaceWeight
minFaceWeight 0.01;

//MinVolRatio
minVolRatio 0.005;

//MinTriabgleTwist
minTriangleTwist -1;

// Advanced

//Number of error distribution iterations
nSmoothScale 4;

//Amount to scale back displacement at error points
errorReduction 0.75;

relaxed
{
    maxNonOrtho 90;
    maxBoundarySkewness 20;
    maxInternalSkewness 4;
}
}
```

ALLEGATO 3

Dizionario *extrudeMeshDict* per il processo di estrusione della *Mesh*

```

/*-----* - C++ - *-----*/
|=====|
| \ \ \ \ / Field | OpenFOAM: The Open Source CFD Toolbox
| \ \ \ \ / Operation | Version: 2.3.0
| \ \ \ \ / And | Web: www.OpenFOAM.org
| \ \ \ \ / Manipulation |
/*-----*-----*/

```

FoamFile

```

{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       extrudeMeshDict
}

```

```

// *****

```

```

constructFrom    mesh;

sourceCase       "./";

sourcePatches    (bottom);

flipNormals      false;

extrudeModel     linearDirection;

nLayers          4;

expansionRatio   1.0;

linearDirectionCoeffs
{
    direction     (0 0 -1);
    thickness      0.037;
}

mergeFaces       false;

```


ALLEGATO 4

Script *createBaseMesh* per la generazione dinamica della *Mesh*

```
#!/bin/sh
cd ${0%/*} || exit 1

# arguments:
if [ ! -n "$1" ]
then
    deg0=0
    echo "Reference angle = 0"
else
    deg0=$1
fi

# Set curvature radius
r=0.01015

# Calculate rotor displacement along "x" and "y"
rad0=$(echo "$deg0/180*3.141592654" | bc -l)
dx=$(echo "$r*c($rad0)" | bc -l)
dy=$(echo "$r*s($rad0)" | bc -l)

# Copy original files to new files
cp -r constant/triSurface/statorBase.stl constant/triSurface/stator.stl
cp -r constant/triSurface/rotorBase.stl constant/triSurface/rotor.stl

# Translate rotor
surfaceTransformPoints -translate "($dx $dy 0)"
constant/triSurface/rotor.stl
constant/triSurface/rotor.stl

# Run "blockMesh" utility
blockMesh

# Extract edges from the geometry surface
surfaceFeatureExtract

# Run "snappyHexMesh" utility overwriting old Mesh
snappyHexMesh -overwrite

# Run "extrudeMesh" utility
extrudeMesh
```


BIBLIOGRAFIA

- [1] H. K. Wersteeg, W. Malalasekera
An introduction to Computational Fluid Dynamics
Longman Scientific & Technical, 1995

- [2] J. H. Ferziger, M. Peric
Computational Methods for Fluid Dynamics
Springer, 2002

- [3] Charles Hirsch
Numerical Computation of Internal & External Flows – Volume 1
Elsevier, 2007

- [4] B. Blunier, G. Cirrincione, A. Miraoui
Novel Geometrical Model of Scroll Compressors for the Analytical Description of the Chamber Volumes
Purdue University e-Pubs, 2006

- [5] J. Marchese
Dynamics of an Orbiting Scroll with Axial Compliance
Purdue University e-Pubs, 1992

- [6] Y. C. Chang, K. Y. Liang, C. F. Lay, Y. Z. R. Hu, G. D. Tarng
Performance Investigation with Assembly Tolerance in Scroll Compressors
Purdue University e-Pubs, 2000

- [7] Yangguang Liu, Yueh-ju Tang, Yu-Choung Chang
R410A Scroll Compressor Design Verification used with Finite Element Analysis
Purdue University e-Pubs, 2012

- [8] James W. Bush, John P. Elson
Scroll Compressor Design Criteria for Residential Air Conditioning and Heat Pump Applications
Purdue University e-Pubs, 1988
- [9] T. Fushiki, F. Sano, K. Ikeda, T. Nishiki, T. Sebata
Development of a Scroll Compressor with New Compliant Mechanism
Purdue University e-Pubs, 2002
- [10] K. Yano, H. Nakao, M. Shimoji
Development of Large Capacity Scroll Compressor
International Compressor Engineering Conference at Purdue, 2008
- [11] Peter Howell
Fluid Mechanical Modelling of the Scroll Compressor
University of Oxford
- [12] F. Yi, Y. Qian
Developing a Compact Automotive Scroll Compressor
Purdue University e-Pubs, 2008
- [13] J. W. Bush, J. Caillat, S. M. Seibel
Dimensional Optimization of Scroll Compressors
Purdue University e-Pubs, 1986
- [14] B. Blunier, G. Cirrincione, Y. Herve, A. Miraoui
A New Analytical and Dynamic Model of a Scroll Compressor with Experimental Validation
International Journal of Refrigeration, 2009

- [15] Z. Jiang, K. Cheng, D. K. Harrison
A Concurrent Engineering Approach to the Development of a Scroll Compressor
Journal of Materials Processing Technology, 2000
- [16] Carrier Corporation
Scroll Compressors – High Efficiency Compression for Commercial and Industrial Applications
2004
- [17] Paul C. Hanlon
Compressor Handbook
McGraw – Hill, 2001
- [18] G. Montenegro, A. Della Torre, M. Fiocco
Simulation of a Rotary Machine with OpenFoam – LibRotaryMachine-2.2.x
Politecnico di Milano – Internal Combustion Engines Group, 2013
- [19] OpenFOAM
User Guide – Version 2.3.0
2014
- [20] OpenFOAM
Programmer’s Guide – Version 2.3.0
2014