

**ALMA MATER STUDIORUM - UNIVERSITÀ DI
BOLOGNA**

SCUOLA DI INGEGNERIA E ARCHITETTURA

Dipartimento di Ingegneria Industriale

Corso di Laurea Triennale in Ingegneria Energetica

TESI DI LAUREA

in

Fondamenti di informatica

**Generazione di griglie computazionali per la
progettazione di componenti di reattori nucleari di IV
generazione refrigerati a sodio**

Candidato:
LUDOVICO SPITILLI

Relatore:
Prof. Ing. SANDRO MANSERVISI

Correlatore:
Ing. ANTONIO CERVONE

**Anno Accademico 2018/2019
II Sessione**

English abstract

Computational thermo-fluid dynamics is a powerful and well-established tool adopted in the development and analysis of components of nuclear reactors. To get accurate results a valid computational model has to be developed, starting from a suitable computational mesh.

In this thesis, we present the development of a mesh for the simulation of one component of the ESFR (European Sodium Fast Reactor), done together with ENEA. The REDAN is a steel liner that divides the hot and cold pools of the reactor, and it is difficult to model in thermal-hydraulic system codes due to its open geometry.

The thesis explores two different approaches to the construction of a suitable geometry and mesh for the hot pool of the reactor, delimited by the REDAN. In the first case, the sections of the geometry that show high regularity are meshed with a structured approach, while the others are kept unstructured. In the second one, all surfaces are meshed with the unstructured algorithm. A comparison of OpenFOAM simulations done with the two different meshes shows that the first approach leads to better results with a comparable computational cost of the simulation.

Indice

Introduzione	5
1 Richiami di Termofluidodinamica	6
1.1 Elementi basilari della Fluidodinamica	6
1.1.1 Descrizione del moto di un fluido	6
1.1.2 Modellizzazione del termine temporale	7
1.1.3 Modellizzazione delle forze	8
1.2 Equazione di continuità	9
1.3 Equazione di bilancio locale della quantità di moto	11
1.3.1 Calcolo delle forze di superficie	11
1.3.2 Calcolo delle forze di volume	12
1.3.3 Equazione di Navier-Stokes	13
1.4 Equazione della Temperatura	13
1.5 La Turbolenza	15
2 Fluidodinamica Computazionale	17
2.1 Modello geometrico e generazione mesh	18
2.2 Il programma di simulazione CFD: OpenFOAM	21
2.3 La visualizzazione e analisi dei dati con Paraview	23
3 Reattori di IV generazione	25
3.1 ESFR: European Sodium Fast Reactor	26
3.1.1 Schema e principi generali di funzionamento	26
3.1.2 ACS: Above Core Structures	27
3.1.3 IHX: Intermediate Heat Exchanger	28
3.1.4 Pompa Primaria	29
3.1.5 Sistema Secondario	30
3.2 Redan	30

4 Creazione del mesh per la simulazione termo-fluidodinamica del REDAN	32
4.1 Analisi della mesh	32
4.2 Divisione del dominio e creazione della geometria	36
4.2.1 Superfici Fisiche	37
4.3 Creazione della Mesh	40
5 Analisi dei risultati	44
5.1 Tipologia di problema e condizioni iniziali	44
5.2 Confronto tra le mesh e campi di soluzione	46
Conclusioni	51
Bibliografia	53

Introduzione

Nel corso degli ultimi 30 anni, la ricerca e lo studio dei reattori nucleari si è orientato verso un approfondimento delle condizioni in cui è possibile produrre energia in estrema sicurezza. A seguito degli incidenti, come *Fukushima* e *Chernobyl*, c'è stata una spinta verso design innovativi con grado di sicurezza maggiore rispetto ai progetti del passato.

La scelta del design dipende da molti fattori, ad esempio la risposta termica di una parete lambita da un fluido, ed è molto importante prevedere il suo comportamento sotto determinate condizioni di lavoro attraverso delle simulazioni.

In questo caso, viene in aiuto la *fluidodinamica computazionale* ovvero l'utilizzo di codici numerici per la risoluzione di problemi di termofluidodinamica con cui è possibile simulare le parti ed i componenti del reattore che si ritengono essere più "delicate" da un punto di vista fisico e che potrebbero rappresentare un problema di sicurezza.

Per ottenere dei risultati accurati bisogna produrre un modello computazionalmente valido di quello di interesse ovvero realizzare una *mesh* adeguata al tipo di simulazione che si vuole realizzare.

Con il presente elaborato di tesi si vuole studiare un caso del genere attraverso la realizzazione della *Mesh*, finalizzata alle simulazioni termofluidodinamiche, di un componente termicamente sollecitato del reattore *ESFR*, un reattore di IV generazione refrigerato al sodio. La mia attività di ricerca è stata svolta nei laboratori dell'ENEA (Ente Nazionale Energie Alternative) e di Montecuccolino.

La struttura della tesi è la seguente. Nel capitolo 1 vengono presentate le equazioni della termofluidodinamica comunemente risolte nelle simulazioni. Nel capitolo 2 vengono introdotti i programmi usati per la realizzazione delle griglie e alcune delle loro funzioni. Nel capitolo 3, dopo un breve accenno ai reattori di IV generazione, verrà descritto l'*ESFR* e il componente oggetto di studio. Nel capitolo 4 verranno presentate le strategie adottate per la realizzazione della griglia. Nel capitolo 5 infine si analizzeranno i risultati ottenuti.

Capitolo 1

Richiami di Termofluidodinamica

Un problema termofluidodinamico incomprimibile si considera definito una volta noto un sistema di equazioni differenziali che sotto opportune ipotesi riesca a descrivere il moto del fluido. Questo significa determinare 5 funzioni scalari di (x, y, z, t) , ovvero velocità (u, v, w) pressione p e temperatura T . Per descrivere il moto sono necessarie due equazioni derivanti da bilanci integrali più una terza per l'energia, ovvero equazione di continuità, equazione di bilancio per la quantità di moto ed equazione della temperatura, formulate sotto l'ipotesi che il fluido sia un *mezzo continuo*: tutti i fenomeni che interessano il moto macroscopico del fluido hanno una scala molto maggiore rispetto a quella microscopica che descrive le dinamiche dei suoi singoli elementi. Per cui il fluido può essere considerato come costituito da un numero infinito di *elementi infinitesimali*, ciascuno contenente un numero di particelle elementari dell'ordine del *numero di Avogadro*, e che evolvono nel tempo lungo una propria traiettoria.

1.1 Elementi basilari della Fluidodinamica

1.1.1 Descrizione del moto di un fluido

Per descrivere il moto di un fluido è necessario definire alcuni strumenti fisico/matematici che risulteranno particolarmente utili nella descrizione delle equazioni sopracitate [1].

Per moto di un fluido si intende il moto dei suoi elementi fondamentali. Supponiamo che un fluido si stia muovendo in una regione di spazio in cui è presente un campo gravitazionale uniforme \mathbf{g} . Ad ogni istante, il moto del fluido può essere visto come la somma di infiniti sistemi chiusi anch'essi in moto. Ciascun sistema chiuso infinitesimo, di volume infinitesimo dV

e velocità \mathbf{u} , sarà detto *Elemento di fluido*. Perciò per Moto del fluido si intenderà il moto degli Elementi di fluido.

Lo stato del fluido può essere stazionario, statico o dipendente dal tempo. Per comportamento statico di un fluido si intende l'analisi con campo di velocità nullo. In un moto stazionario la velocità, in un qualsiasi punto fissato, non dipende dal tempo, ovvero il campo di velocità può essere descritto mediante una funzione vettoriale dello spazio indipendente dal tempo: $\mathbf{u} = \mathbf{u}(x, y, z)$. Il sistema oggetto di studio può essere descritto in due modi diversi: Lagrangiano ed Euleriano. Nel punto di vista Lagrangiano si considera l'osservatore in moto con l'elemento di fluido o porzione di fluido con massa finita. Nell'analisi Euleriana si considera l'osservatore fisso e si analizza un volume di controllo, di estensione infinitesima o finita.

La modellizzazione del moto viene descritta attraverso equazioni a derivate spaziali. Nel seguito daremo la descrizione dei vari termini dell'equazione a cominciare dal termine contenente la derivata Lagrangiana.

1.1.2 Modellizzazione del termine temporale

Per introdurre il concetto di *derivata Lagrangiana* si utilizzi l'osservazione del moto da un punto di vista Lagrangiano e si scelga come sistema un elemento di fluido infinitesimo, δA , che si muove lungo la sua traiettoria. Allora ogni proprietà scalare f di δA (Temperatura, pressione, componente della velocità) varia nel tempo mentre δA si sposta lungo la sua traiettoria. Se si dovesse studiare solo la particella δA , f sarebbe una funzione del tempo e la derivata di f rispetto al tempo sarebbe la derivata ordinaria, come nella meccanica del punto materiale. In Fluidodinamica però bisogna studiare infinite particelle come $\delta A \implies f = f(x, y, z, t)$. La derivata rispetto al tempo che si ha seguendo la particella nel suo moto è chiamata derivata sostanziale della funzione f ed è indicata con $\frac{Df}{Dt}$ definita come

$$\frac{Df}{Dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta f}{\Delta t}. \quad (1.1)$$

Essendo che

$$\Delta f \approx \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial z} \Delta z + \frac{\partial f}{\partial t} \Delta t,$$

otteniamo

$$\frac{\Delta f}{\Delta t} \approx \frac{\partial f}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial f}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial f}{\partial z} \frac{\Delta z}{\Delta t} + \frac{\partial f}{\partial t}.$$

Poiché la variazione Δf dev'essere calcolata seguendo il moto della particella, si avrà, per $\Delta t \rightarrow 0$

$$\frac{\Delta x}{\Delta t} \rightarrow u, \quad \frac{\Delta y}{\Delta t} \rightarrow v, \quad \frac{\Delta z}{\Delta t} \rightarrow w.$$

Quindi la (1.1) diventa

$$\frac{Df}{Dt} = \frac{\partial f}{\partial x}u + \frac{\partial f}{\partial y}v + \frac{\partial f}{\partial z}w + \frac{\partial f}{\partial t}, \quad (1.2)$$

scrivibile nella forma

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + (\mathbf{u} \cdot \nabla)f. \quad (1.3)$$

1.1.3 Modellizzazione delle forze

Le forze agenti su un elemento di fluido possono essere di due tipi: le forze di superficie e quelle di massa.

Le forze di massa sono le forze che agiscono direttamente all'interno dell'elemento di fluido, dovute a campi di forza esterni (ad esempio la forza dovuta ad un campo gravitazionale $\mathbf{g} = (0, 0, -g)$).

Le forze di superficie sono forze distribuite sulla sua superficie, esercitate dagli elementi di fluido adiacenti. La forza per unità di area che descrive questa interazione è lo sforzo \mathbf{q} che, a sua volta, può essere calcolato mediante il tensore delle tensioni σ .

Per introdurre il concetto di forza superficiale introduciamo un oggetto matematico detto il *tensore delle tensioni*. Per fare questo si consideri una porzione di fluido delimitato da una superficie chiusa S come in figura 1.1. Il fluido esterno ad S esercita su quello interno, in ogni posizione \mathbf{P} , una forza per unità di superficie \mathbf{q} , detta sforzo. Per ogni punto \mathbf{P} di S , \mathbf{q} dipende soltanto dalla direzione $\hat{\mathbf{n}}$ della normale ad S in \mathbf{P} ; inoltre le componenti di \mathbf{q} sono funzioni lineari omogenee delle componenti di $\hat{\mathbf{n}}$. Come conseguenza esiste una matrice

$$\sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix},$$

tale che $\mathbf{q} = \sigma^t \hat{\mathbf{n}}$

$$\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}. \quad (1.4)$$

La matrice σ è chiamata *Tensore delle tensioni*, gli elementi sulla diagonale principale tensioni *normali* mentre quelli fuori diagonale tensioni *tangenziali*; si può dimostrare che σ è simmetrica, ovvero che $\sigma^t = \sigma$. Indicando con \mathbf{c}_i l' i -esimo vettore colonna di σ , dalla (1.4) si ottiene

$$\mathbf{q}_i = \mathbf{c}_i \cdot \hat{\mathbf{n}}.$$

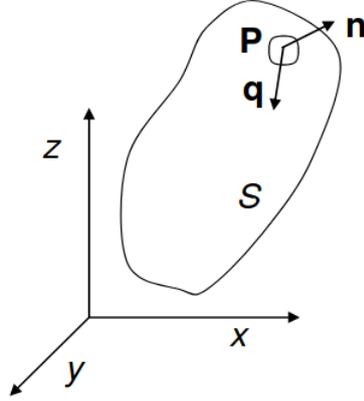


Figura 1.1: Sforzo e direzione della normale sulla superficie S

La matrice precedente, risulta utile nella definizione della pressione p di un fluido in un punto. Chiamando con $Tr(\sigma)$ la Traccia della matrice σ , ovvero la somma degli elementi sulla diagonale principale, allora

$$p = -\frac{1}{3} Tr(\sigma) = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}). \quad (1.5)$$

Così facendo è possibile scomporre la matrice σ nella somma di due matrici

$$\sigma = \begin{pmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{pmatrix} + \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix}, \quad (1.6)$$

dove il tensore τ è chiamato *Tensore degli sforzi viscosi*. La relazione fra gli elementi del tensore σ e τ è

$$\sigma_{ij} = \tau_{ij} - pI_{ij}, \quad (1.7)$$

dove I_{ij} è il generico elemento della matrice identità.

1.2 Equazione di continuità

Si consideri un volume di controllo V che occupa la regione di spazio Θ , delimitata da una superficie chiusa S . Concettualmente, il principio di conservazione della massa è definito da

$$\frac{\text{variazione massa contenuta in } V \text{ in } dt}{dt} = \frac{\text{massa entrante in } V \text{ in } dt}{dt} \quad (1.8)$$

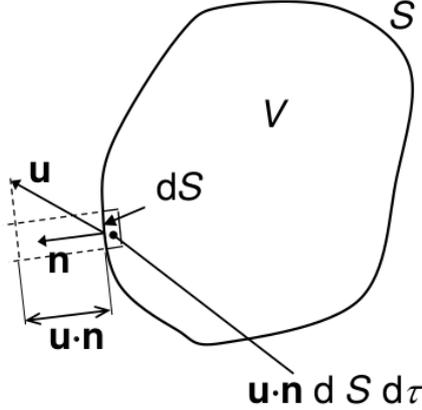


Figura 1.2: Volume di controllo \$V\$ e superficie \$S\$

Indicando con \$\rho\$ la densità del fluido, il primo membro della (1.8) è esprimibile come

$$\frac{d}{dt} \int_V \rho dV = \int_V \frac{\partial \rho}{\partial t} dV, \quad (1.9)$$

dove si è potuto portare la derivata (parziale perché \$\rho = \rho(x, y, z, t)\$) entro l'integrale perché il dominio di integrazione \$V\$ non dipende dal tempo.

Si consideri un elemento infinitesimo \$dS\$ della superficie \$S\$, nell'intorno di un punto \$P\$. Sia \$\hat{\mathbf{n}}\$ il versore normale alla superficie \$S\$ in \$P\$, e sia \$\mathbf{u}\$ il vettore velocità del fluido in \$P\$. Un elemento di fluido posto in \$P\$ compie, nel tempo \$dt\$, uno spostamento la cui componente nella direzione di \$\hat{\mathbf{n}}\$ è: \$\mathbf{u} \cdot \hat{\mathbf{n}} dt\$. Il volume di fluido che esce da \$dS\$ nel tempo \$dt\$ è invece \$\mathbf{u} \cdot \hat{\mathbf{n}} dS dt\$. Di conseguenza il secondo membro dell'uguaglianza (1.8) diventa

$$- \int_S \rho \mathbf{u} \cdot \hat{\mathbf{n}} dS, \quad (1.10)$$

dove il segno \$-\$ è stato introdotto poiché nella (1.8) la massa è entrante, mentre \$\hat{\mathbf{n}}\$ è uscente (si veda la Figura 1.2). Uguagliando la (1.9) e la (1.10) si ottiene

$$\int_V \frac{\partial \rho}{\partial t} dV = - \int_S \rho \mathbf{u} \cdot \hat{\mathbf{n}} dS. \quad (1.11)$$

Se si applica il teorema di Gauss-Green alla (1.10) e si inserisce nella (1.11) si ottiene la seguente equazione

$$\int_V \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) dV = 0. \quad (1.12)$$

La (1.12) vale per qualsiasi volume V attraversato da un fluido in moto, perciò dev'essere

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1.13)$$

Nel caso particolare di ρ costante, la (1.13) assume la forma

$$\nabla \cdot \mathbf{u} = 0, \quad (1.14)$$

che esplicitata nelle 3 direzioni (x,y,z) diventa

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0.$$

1.3 Equazione di bilancio locale della quantità di moto

Si consideri un elemento di fluido ΔA che ha un volume elementare ΔV e una massa elementare $\Delta m = \rho \cdot \Delta V$. Se applichiamo la seconda legge della dinamica, la somma di tutte le forze agenti sul fluido dev'essere pari al prodotto della massa Δm per l'accelerazione, che in fluidodinamica è espressa come derivata sostanziale della velocità $\hat{\mathbf{u}}$. Indicando con

$\Delta \mathbf{f}_b$: risultante delle forze di volume

$\Delta \mathbf{f}_s$: risultante delle forze di superficie

$\Delta \mathbf{f}$: risultante delle forze

si ottiene l'equazione

$$\rho \Delta V \frac{D\mathbf{u}}{Dt} = \Delta \mathbf{f} = \Delta \mathbf{f}_b + \Delta \mathbf{f}_s. \quad (1.15)$$

1.3.1 Calcolo delle forze di superficie

Ricordando la sezione sul *Tensore delle Tensioni* sezione 1.1.3, è possibile scrivere

$$\Delta f_{si} = \int_{\Delta S} q_i dS = \int_{\Delta S} (\mathbf{c}_i \cdot \hat{\mathbf{n}}) dS.$$

Poiché ΔS è una superficie chiusa, applicando il teorema di Gauss-Green l'equazione diventa

$$\Delta f_{si} = \int_{\Delta V} (\nabla \cdot \mathbf{c}_i) dV. \quad (1.16)$$

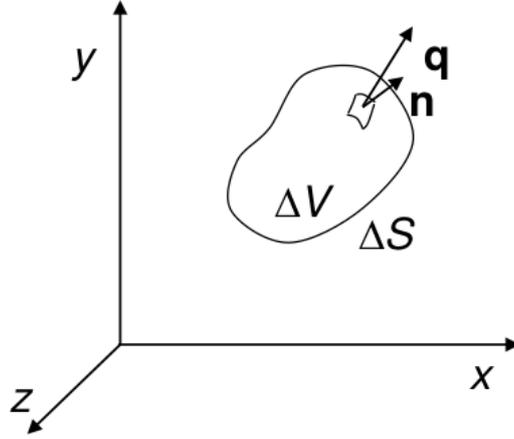


Figura 1.3: Volume di controllo V e superficie S

Poiché la regione di spazio ΔV è infinitamente piccola, in essa si può considerare $(\nabla \cdot \mathbf{c}_i)$ costante e la (1.16) diventa

$$\Delta f_{si} = (\nabla \cdot \mathbf{c}_i) \Delta V = \left(\sum_{j=1}^3 \frac{\partial \sigma_{ji}}{\partial x_j} \right) \Delta V. \quad (1.17)$$

Riprendendo la (1.7), la (1.17) diventa

$$\Delta f_{si} = \left(-\frac{\partial p}{\partial x_j} + \sum_{j=1}^3 \frac{\partial \tau_{ji}}{\partial x_j} \right) \Delta V. \quad (1.18)$$

La (1.18) dà l'espressione generale della risultante delle forze di superficie

$$\Delta \mathbf{f}_s = (-\nabla p + \nabla \cdot \boldsymbol{\tau}) \Delta V.$$

1.3.2 Calcolo delle forze di volume

Si supponga che l'unica forza di volume sia un campo gravitazionale uniforme, di potenziale $\mathbf{g}z = (0, 0, gz)$. Si ha allora

$$\Delta \mathbf{f}_b = -\rho \Delta V \nabla(\mathbf{g}z). \quad (1.19)$$

Introducendo la (1.18) e la (1.19) nella (1.15), e dividendo per ΔV si ottiene

$$\rho \frac{D\mathbf{u}}{Dt} = -\rho \nabla(\mathbf{g}z) - \nabla p + \nabla \cdot \boldsymbol{\tau}. \quad (1.20)$$

La equazione (1.20) è chiamata *equazione vettoriale di bilancio locale della quantità di moto* e vale per qualsiasi fluido, con le sole ipotesi che possa essere utilizzata la meccanica dei continui e che l'unico campo di forza sia un campo gravitazionale uniforme.

1.3.3 Equazione di Navier-Stokes

Quando nel fluido il vettore velocità \mathbf{u} è non uniforme, il tensore degli sforzi viscosi τ non è ovunque nullo. Se il fluido è delimitato da una superficie solida, a contatto con la parete il fluido è fermo (rispetto alla parete) e si hanno forti gradienti di \mathbf{u} . Questi gradienti vengono descritti mediante il tensore delle deformazioni D , nella forma

$$D_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

Il legame fra gli elementi del tensore τ e gli elementi del tensore D viene descritto attraverso una grandezza chiamata *viscosità dinamica del fluido*: μ , tale da poter esprimere τ come

$$\tau_{ij} = 2\mu D_{ij} - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})I_{ij}. \quad (1.21)$$

Inserendo nella (1.20) la (1.21) e sviluppando i calcoli si ottiene l'equazione

$$\rho \frac{D\mathbf{u}}{Dt} = -\rho \nabla(\mathbf{g}z) - \nabla p + \mu \left[\nabla^2 \mathbf{u} + \frac{1}{3} \nabla(\nabla \cdot \mathbf{u}) \right]. \quad (1.22)$$

più comunemente chiamata *Equazione di Navier-Stokes*, che corrisponde a 3 equazioni scalari nelle incognite (x, y, z) .

1.4 Equazione della Temperatura

Consideriamo un elemento di fluido di massa dm immerso in un campo gravitazionale g ad un'altezza z . L'energia E per unità di massa, ψ , è esprimibile come

$$\psi = e + \frac{\mathbf{u}^2}{2} + gz, \quad (1.23)$$

ovvero come somma di *energia interna*, *energia cinetica* ed *energia potenziale* tutte per unità di massa. Il bilancio termodinamico dell'energia, in generale, può essere scritto come

$$\frac{\partial E}{\partial t} = Q + W, \quad (1.24)$$

dove con Q si è indicata la *potenza termica* e con W la potenza meccanica ricevuta dall'elemento di fluido. A sua volta la potenza termica può essere espressa come somma di due contributi, uno che tiene conto della potenza termica che attraversa la superficie dell'elemento Q_s , e uno che tiene conto della potenza termica generata per unità di volume Q_g . Il primo può essere espresso per unità di superficie, attraverso il vettore \mathbf{q} , il secondo per unità di volume, introducendo q_g . Per quanto riguarda la potenza meccanica, essa coincide con il lavoro per unità di tempo prodotto dalle azioni di trazione sulla superficie dell'elemento di fluido, e dipenderà dunque dal tensore delle tensioni σ (si veda l'equazione (1.4)). Ricordando la definizione di *derivata sostanziale*, si può ottenere l'equazione di conservazione dell'energia come

$$\rho \frac{D}{Dt} \left(e + \frac{\mathbf{u}^2}{2} + gz \right) = -\nabla \cdot \mathbf{q} - \nabla \cdot p\mathbf{u} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u}) + Q_g. \quad (1.25)$$

Sviluppando in termini, la (1.25) diventa

$$\rho \frac{D}{Dt} \left(e + \frac{\mathbf{u}^2}{2} + gz \right) = -\nabla \cdot \mathbf{q} - p\nabla \cdot \mathbf{u} - \mathbf{u} \cdot \nabla p + \mathbf{u} \cdot (\nabla \cdot \boldsymbol{\tau}) + \boldsymbol{\tau} : \nabla \mathbf{u} + Q_g. \quad (1.26)$$

E' possibile inoltre ottenere l'*equazione di conservazione dell'energia meccanica* moltiplicando la (1.20) scalarmente per \mathbf{u} ovvero

$$\rho \frac{D\mathbf{u}}{Dt} = -\mathbf{u} \cdot \rho \nabla(gz) - \mathbf{u} \cdot \nabla p + \mathbf{u} \cdot (\nabla \cdot \boldsymbol{\tau}), \quad (1.27)$$

che può essere riformulata come

$$\rho \frac{D}{Dt} \left(\frac{1}{2} \mathbf{u}^2 + gz \right) = -\mathbf{u} \cdot \nabla p + \mathbf{u} \cdot (\nabla \cdot \boldsymbol{\tau}). \quad (1.28)$$

All'interno della parentesi a primo membro compaiono la derivata sostanziale dell'energia cinetica e potenziale e si sono messe in relazione le variazioni di energia meccanica con l'azione esercitata dalle forze agenti sulla superficie del volume di controllo. Se si sottrae alla (1.25) la (1.28) si ottiene l'*equazione di conservazione dell'energia interna*

$$\rho \frac{De}{Dt} = -\nabla \cdot \mathbf{q} - p\nabla \cdot \mathbf{u} + \boldsymbol{\tau} : \nabla \mathbf{u} + Q_g, \quad (1.29)$$

che permette di calcolare le variazioni di temperatura dovute a

$\nabla \cdot \mathbf{q}$ trasmissione di calore per conduzione

$p\nabla \cdot \mathbf{u}$ compressione reversibile

$\tau : \nabla \mathbf{u}$ dissipazione viscosa irreversibile

Q_g generazione interna .

E' possibile trasformare l'equazione (1.29) in una nella variabile T . Per farlo bisogna introdurre la *legge di Fourier* che lega il flusso termico \mathbf{q} alla temperatura T attraverso la conducibilità termica k (funzione della temperatura), ovvero

$$\mathbf{q} = -k\nabla T. \quad (1.30)$$

Sotto l'ipotesi di fluido newtoniano è possibile definire la funzione di dissipazione viscosa come

$$\Phi = \tau : \nabla \mathbf{u}. \quad (1.31)$$

Se il fluido è approssimabile come incomprimibile, allora l'energia interna e è funzione solamente della temperatura ed è legata al calore specifico del fluido c nella forma

$$de = cdT. \quad (1.32)$$

Sostituendo la (1.30), la (1.31) e la (1.32) nella (1.29) si ottiene l'*equazione della Temperatura* ovvero

$$c\rho \frac{DT}{Dt} = \nabla \cdot (k\nabla T) + \Phi + Q_g. \quad (1.33)$$

1.5 La Turbolenza

Le equazioni sopracitate hanno validità generale e possono essere applicate a qualsiasi caso si voglia; tuttavia è necessario fare una distinzione tra flusso laminare e flusso turbolento.

In *regime laminare* il sistema si comporta come se fosse composto da filetti fluidi adiacenti che scorrono l'uno sull'altro, non generando alcun tipo di rimescolamento all'interno del fluido. In tale regime le eventuali piccole perturbazioni tendono a smorzarsi. In *regime turbolento*, invece, le perturbazioni tendono a intensificarsi, rendendo il moto irregolare e fortemente dipendente dalle condizioni al contorno. La distinzione tra i due regimi avviene attraverso un parametro adimensionale chiamato *Numero di Reynolds* definito come

$$\text{Re} = \frac{UL}{\nu} \quad (1.34)$$

dove

$$\nu = \frac{\mu}{\rho},$$

è chiamata *viscosità cinematica*. Il regime turbolento si manifesta per elevati valori del numero di Reynolds, quindi per elevate velocità U o quando la viscosità cinematica del fluido in esame è molto bassa.

Capitolo 2

Fluidodinamica Computazionale

La *fluidodinamica computazionale* è l'analisi dei sistemi che trattano movimenti fluidi, scambio di calore e reazioni chimiche, attraverso l'uso di simulazioni effettuate col computer. Le analisi dei risultati ottenuti attraverso simulazioni termofluidodinamiche permettono di prevedere le conseguenze di un particolare fenomeno. Se si considera ad esempio un flusso d'acqua e si inserisce al suo interno un ostacolo, fissati alcuni parametri, ovvero le *condizioni iniziali* e le *condizioni al bordo*, è possibile conoscere la velocità in ogni punto del flusso d'acqua, attraverso la risoluzione numerica delle *equazioni di Navier-Stokes*. I risultati delle simulazioni cambiano in funzione del tipo di modello utilizzato e della potenza computazionale che si è disposti ad impiegare.

I modelli più utilizzati sono:

Direct Numerical Simulation (DNS): le equazioni di Navier-Stokes vengono risolte senza utilizzare alcun modello di Turbolenza. La griglia computazionale deve essere fitta abbastanza da poter riprodurre tutte le scale di vortici presenti nel fluido, e quindi il costo computazionale è molto elevato, anche per bassi numeri di Re .

Large Eddy Simulation (LES): riduce il costo computazionale “filtrando” le equazioni di Navier-Stokes, ovvero rimuovendo le soluzioni di piccola scala attraverso una media temporale e spaziale. Il costo computazionale è minore rispetto al precedente.

Reynolds Averaged Navier-Stokes (RANS): le grandezze in gioco non sono più istantanee ma mediate in un certo periodo di tempo sufficientemente piccolo rispetto a quello del fenomeno da seguire. Questo consente una notevole riduzione del tempo di calcolo, poiché le scale del moto *medio* sono più grandi di quelle del moto turbolento. In questa

maniera, però, tutte le informazioni sulla piccola scala sono perse nel modello.

2.1 Modello geometrico e generazione mesh

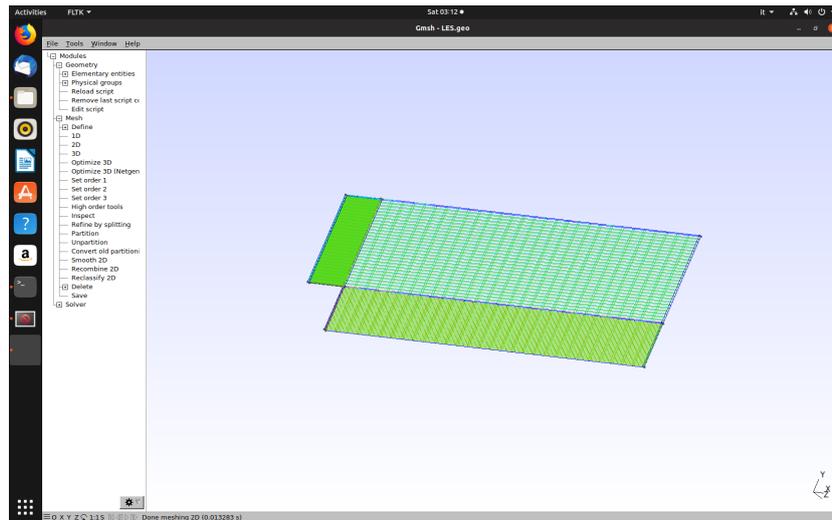


Figura 2.1: Interfaccia grafica di Gmsh con semplice geometria

Prima di affrontare una simulazione, bisogna creare una geometria computazionalmente “valida” (leggibile ed interpretabile numericamente dal simulatore) del fenomeno considerato. Una volta nota la geometria, è necessario scomporla in una fitta rete di celle elementari, sulle quali vengono risolte le equazioni di conservazione. Questo processo è chiamato creazione della *mesh* o anche *meshing*.

La qualità della mesh ha un impatto considerevole sull’accuratezza della soluzione e sul tempo di calcolo e può essere considerata il punto di partenza di ogni simulazione.

Gmsh è un programma open-source che fornisce all’utente una serie di strumenti per la creazione delle mesh attraverso una interfaccia grafica user-friendly [2]. Il funzionamento si basa su 4 moduli, tutti controllabili dalla GUI: Geometry, Mesh, Solver, Post-Processing. Può essere avviato, dopo l’installazione, direttamente dal terminale con il comando: `gmsh file.geo` dove `geo` è l’estensione dei file gestiti da Gmsh. Una volta dato il comando, l’ambiente si presenta come mostrato in Figura 2.1. La creazione della geometria può avvenire in due modi: direttamente dalla GUI attraverso il linguaggio di programmazione di Gmsh.

Il primo metodo è dinamico, ovvero si visualizzano direttamente a schermo tutte le operazioni (creazione punti, linee, superfici, ...). Il secondo metodo prevede la creazione di un file `.geo` in cui la geometria viene realizzata con il linguaggio di programmazione, che associa ad ogni linea, punto ecc. un determinato codice. Tra questi troviamo

Punto : $P_i = \{x, y, z, lcar\}$;

Linea : $Line_i = \{P_i, P_{i+1}\}$;

Loop : $Line Loop_i = \{Line_i, Line_{i+1}, Line_{i+2}, Line_{i+3}\}$.

Il linguaggio nativo di Gmsh, presenta alcuni punti deboli non indifferenti come la numerazione in ordine di punti e linee, che in caso di geometrie molto complesse può portare gli indici ad essere complicati da gestire.

Per questo motivo è stata implementata una versione in Python di Gmsh nota come *PyGmsh*, con il quale è stata costruita la mesh presentata nell'elaborato di tesi [3]. In questa versione, ogni comando di Gmsh è scritto in modo da essere leggibile da Python, ad esempio

Punto : `Pointi=geom.add_point(x,y,z,lcar);`

Linea : `Linei=geom.add_line(Pi, Pi+1);`

Loop : `Loopi=geom.add_line_loop(Linei, Linei+1, Linei+2, Linei+3);`

Una volta eseguito con il comando `python3 file.py` viene creato l'eseguibile `.geo`, che può essere quindi letto da Gmsh. Esistono diversi tipi di celle

Dimensione	Elemento
2D	Triangolari
2D	Rettangolari
3D	Tetraedriche
3D	Piramidali
3D	Esaedriche

Tabella 2.1: Diversi elementi

in cui può essere scomposto un dominio, riportati in Tabella 2.1. Definendo *griglia* la fitta rete di celle elementari, allora a seconda dell'organizzazione delle celle si possono distinguere

Griglie *Strutturate*. Ogni cella è identificata da una terna (i, j, k) all'interno del dominio. Le celle sono organizzate in maniera coerente e le linee della griglia attraversano tutto il dominio. Possibile solo per geometrie relativamente semplici (Figura 2.3).

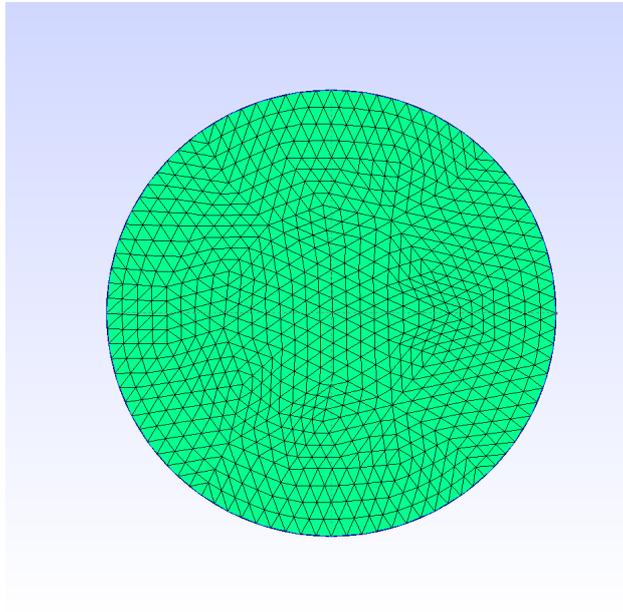


Figura 2.2: Griglia non strutturata

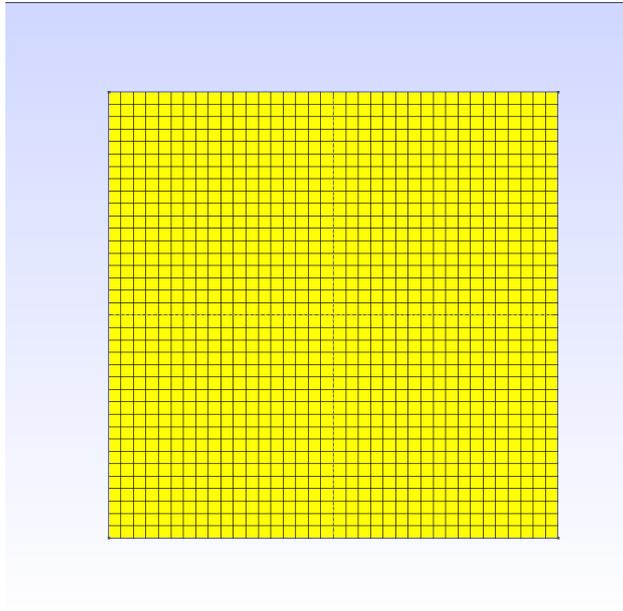


Figura 2.3: Griglia strutturata

Griglie *Non Strutturate*. Non è definita una terna (i, j, k) , non ci sono limiti nel modo in cui le celle sono organizzate ed è richiesto un in-

cremento di memoria e di tempo per il calcolo computazionale (Figura 2.2).

La realizzazione di griglie strutturate è possibile soltanto per geometrie semplici. Le geometrie più complesse, come quella presentata in questo elaborato, presenteranno una griglia mista, ovvero strutturata, dove è possibile farlo, e non strutturata, ma con alto grado di finezza, dove non si può.

2.2 Il programma di simulazione CFD: OpenFOAM

Una volta che è stata creata la Geometria, è necessario risolvere le equazioni della fluidodinamica all'interno del nostro dominio. Per farlo c'è bisogno di un risolutore, che in questo caso è rappresentato da *OpenFOAM* [4].

OpenFoam è un software gratis, open-source che viene aggiornato ogni 6 mesi e spazia dalla fluidodinamica, alla trasmissione del calore all'elettrodinamica. Per poter affrontare una simulazione, è necessario creare un file, nel linguaggio di OpenFOAM, più appropriato per il caso considerato, oppure è possibile usare i risolutori standard presenti nel programma nella sezione `tutorial`

Comunque si scelga di procedere, per iniziare una simulazione, è necessario creare una cartella in cui devono essere presenti alcuni file contenuti in opportuni `folder` (Figura 2.4)

`0` : All'interno vengono contenuti i file relativi alle condizioni iniziali per il fenomeno considerato come velocità, pressione e temperatura.

`constant` : Tiene conto delle proprietà fisiche e geometriche del dominio ed è composta da

`polyMesh` : Viene specificata la descrizione della mesh

`caseProperties` : Contiene i parametri propri della simulazione `case`. In una simulazione fluidodinamica ad esempio avremmo avuto `transportProperties` : che contiene il valore della viscosità del fluido.

`system` : Contiene files per settare parametri inerenti alla soluzione della simulazione.

`controlDict` : Permette di scegliere l'estensione temporale della simulazione e ogni quanto scrivere i dati;

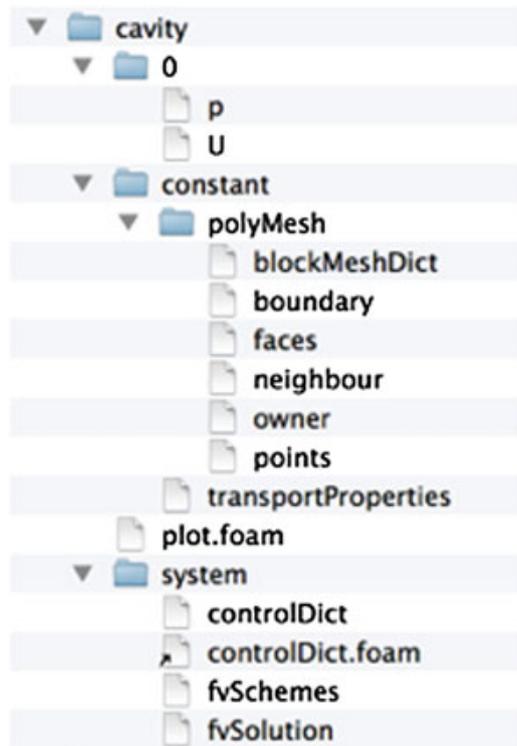


Figura 2.4: Directory di OpenFOAM per il caso di flusso entro una cavità.

`blockMeshDict` : Descrizione dei parametri per la generazione della mesh;

`fvSchemes` : Opzioni relative allo schema a volumi finiti adottato;

`fvSolution` : Opzioni relative alle varie variabili per cui verrà risolto il sistema;

Una volta fissate le condizioni iniziali in 0, bisogna far riconoscere la griglia del modello al risolutore. Per farlo si estrae la mesh nel formato `.msh`, si mette nello stesso folder del caso considerato, e si lancia dal terminale il comando `gmshToFoam`; dopo questo passo è possibile lanciare la simulazione e ottenere i dati che si desiderano dal modello, conservate in cartelle denominate in funzione dell'istante temporale.

La validità dei dati estraibili dalla simulazione dipende dalla qualità della mesh, dall'accuratezza e dalla scelta delle condizioni al contorno e dal risolutore. Inoltre i dati devono avere un significato fisico coerente con il fenomeno che si sta studiando.

2.3 La visualizzazione e analisi dei dati con Paraview



Figura 2.5: Dati provenienti dalla simulazione con un certo intervallo temporale di scrittura.

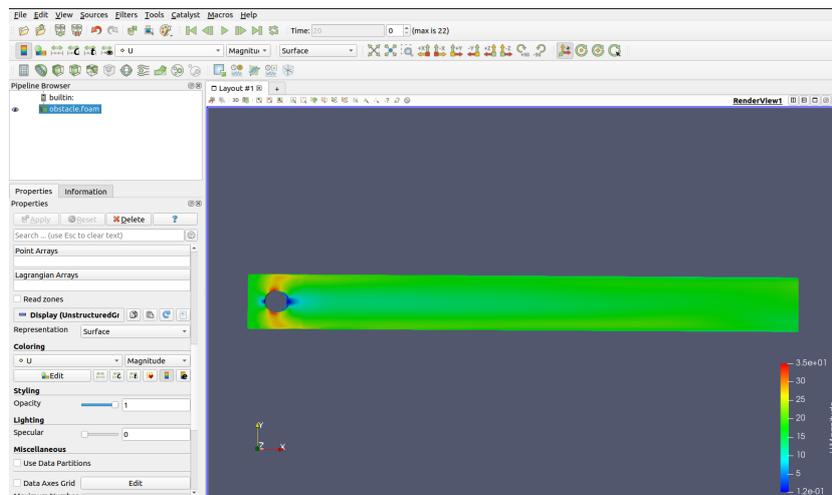


Figura 2.6: Esempio di analisi della velocità in un profilo con ostacolo; colori diversi indicano intensità diverse.

Riuscire a visualizzare i dati ottenuti è di fondamentale importanza ai fini della simulazione: permette ad esempio di studiare l'evoluzione temporale di un determinato fenomeno oppure di visualizzare la parte termicamente più sollecitata di un motore. Questo è possibile grazie a *Paraview*, un software open-source per l'analisi dei dati [5].

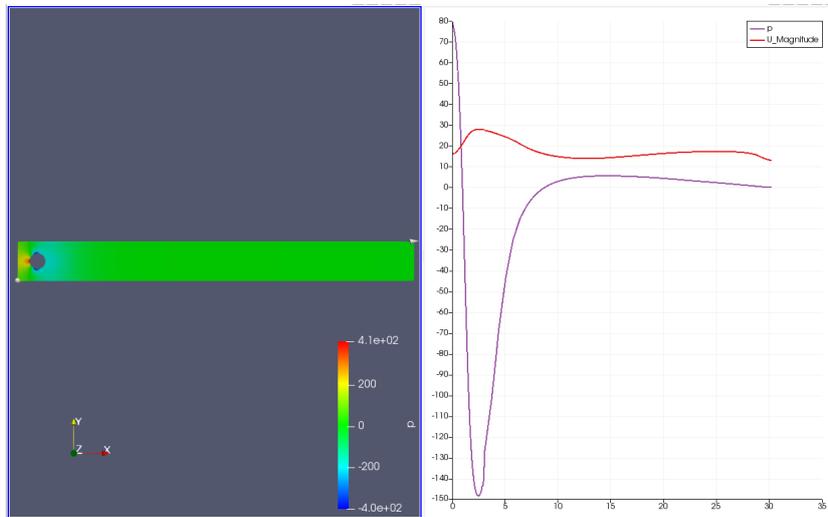


Figura 2.7: Plotting della velocità e della pressione ad un certo istante temporale tra due punti.

All'interno della cartella di *OpenFOAM*, i dati si presentano come riportato in Figura 2.5; fissato il tempo in cui la simulazione deve evolversi e fissato il passo con cui salvare i dati, viene generata una cartella per ogni istante temporale, in cui sono contenuti i valori delle variabili oggetto di studio. Successivamente le cartelle vengono lette dal programma e caricate in modo da visualizzare la geometria e la sua evoluzione.

L'ambiente di *Paraview* si presenta come in Figura 2.6 e sfrutta una organizzazione delle geometrie e dei filtri a pipeline ovvero visualizzati in cascata. Le funzionalità principali sono rappresentate dai tasti Play, Clip, Slice, Glyph, Threshold, Streamtracer e PlotOverLine posti nella parte superiore della finestra. Il tasto Play dà il via all'evoluzione temporale del fenomeno considerato. L'opzione Clip seziona la geometria con un piano e visualizza le variabili dal piano in poi mentre il tasto Slice seziona la geometria con un piano e visualizza le variabili SOLO sul piano. L'opzione Glyph visualizza le variabili assegnandoli un vettore e quella Threshold le celle della mesh che hanno soltanto un determinato valore. L'opzione Streamtracer permette, fissati due punti, visualizzare le linee di flusso di una variabile tra quest'ultimi. L'opzione PlotOverLine, fissati due punti, crea un grafico in cui plotta le variabili in funzione della distanza tra quest'ultimi (Figura 2.7)

Inoltre Paraview permette l'estrazione dei dati in formato Excel e/o testuale così da poterli condividere e maneggiare liberamente.

Capitolo 3

Reattori di IV generazione

I reattori di IV generazione sono il frutto della continua ricerca nell'ambito della sicurezza nucleare, con l'obiettivo di limitare l'esposizione del pubblico, del personale e dell'ambiente agli incidenti. Infatti, sebbene il verificarsi di un incidente sia molto raro, si mira a ridurre la quantità di radiazioni rilasciata e renderla tale da non necessitare ulteriori interventi (confinamento della popolazione, evacuazione).

Il design dei reattori innovativi di IV generazione è pensato sulla base dei precedenti incidenti; si prenda come esempio *Fukushima*: le ricerche svolte negli anni successivi al disastro, hanno fornito una serie di raccomandazioni da mettere in conto nel caso di pericoli naturali, come una significativa autonomia dell'impianto riguardo il tempo necessario per un possibile intervento esterno e l'implementazione di misure di intervento interne/esterne in situ nel caso fosse danneggiato.

Sinteticamente, gli obiettivi da raggiungere sono: sostenibilità, bassi costi, sicurezza, affidabilità e protezione [6]. Per quanto riguarda la sostenibilità verrà generata energia promuovendo l'utilizzo di combustibile a lungo termine, minimizzandone le scorie e il tempo di stoccaggio, così da incidere il meno possibile sull'ambiente e sulla salute. In termini economici, il ciclo di vita dei reattori di IV generazione dovrà essere vantaggioso rispetto ai sistemi di produzione di energia convenzionali, nonché avere un rischio finanziario comparabile con quest'ultimi.

Per quanto riguarda la sicurezza e affidabilità possiamo sottolineare che la produzione di energia nella IV generazione, eccellerà per sicurezza e affidabilità con una bassissima probabilità di danneggiamento del nocciolo. Nella linea per la protezione del materiale nucleare da altri utilizzi ricordiamo che il ciclo di vita del combustibile risulterà meno appetibile per eventuali utilizzi bellici [6].

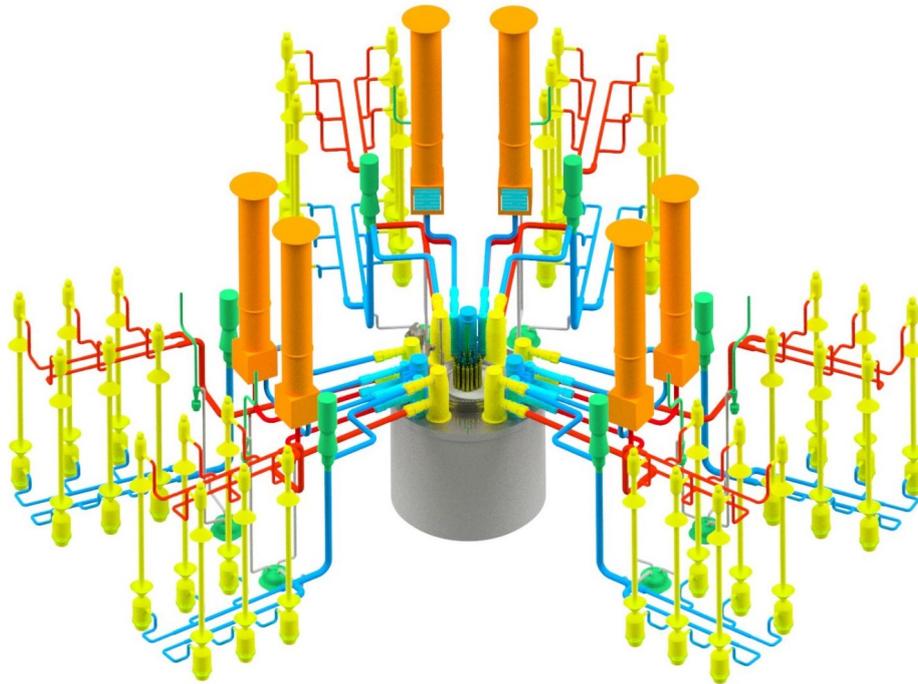


Figura 3.1: Schema generale del reattore e dei sistemi di sicurezza

3.1 ESFR: European Sodium Fast Reactor

Il progetto *ESFR-SMART* (ESFR: European Sodium-cooled Fast Reactor, SMART: Safety Measures Assessment and Research Tools [7]) è un progetto dell'Unione Europea lanciato nel Settembre del 2017 e si propone l'obiettivo di semplificare il design e combinare la sicurezza con le principali caratteristiche dei reattori *SFR*, ovvero: l'efficienza della convenzione naturale, l'alta inerzia termica e lunghi periodi di funzionamento prima dell'intervento umano.

3.1.1 Schema e principi generali di funzionamento

Il reattore, si vedano le figure 3.1 e 3.2, consiste in due piscine, primaria e secondaria, contenenti sodio a diverse temperature. Il sodio (primario perché scorre all'interno del reattore) viene prelevato a 395°C dalla piscina più esterna attraverso 3 *Pompe Primarie (PP)*, ognuna con due ingressi, spingendolo verso il core del reattore. Qui, viene fatto passare attraverso le barre di combustibile, che aumentano la temperatura a 545°C , e l'*ACS* (Above Core Structures), che guida le barre di controllo e riduce la velocità al di so-

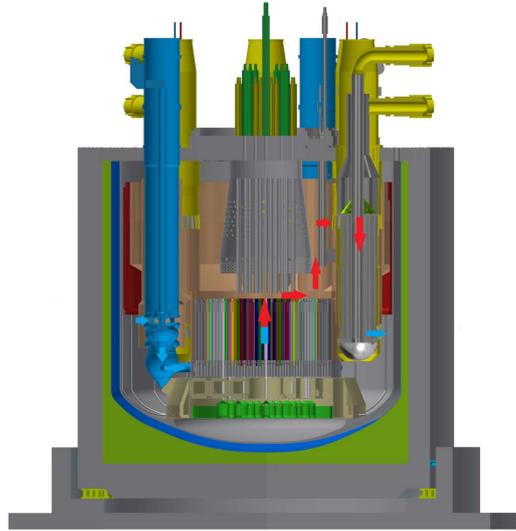


Figura 3.2: Visualizzazione del reattore e del percorso del sodio nel circuito primario. Le frecce rosse e blu indicano un aumento/diminuzione di temperatura nonché il percorso seguito

pra del core. Giunto nella piscina secondaria, il sodio caldo entra in un loop attraverso 6 scambiatori di calore intermedi (*IHX*), nei quali sodio primario e secondario scambiano calore senza entrare in contatto. I fluidi prendono successivamente due strade diverse: il sodio primario, torna nella piscina più esterna, mentre quello secondario, attraverso un nuovo scambiatore di calore (denominato *Steam Generator*) entra in un altro loop, esterno al reattore, in cui scambia calore con l'acqua che, inizialmente a 240°C , raggiunge i 490°C vaporizzando. Il vapore viene successivamente utilizzato per muovere una turbina e produrre, in via teorica, 1500 MWe.

3.1.2 ACS: Above Core Structures

L'*ACS*, si veda figura 3.3, funziona da supporto per il meccanismo che guida le barre di controllo e controlla la distribuzione del flusso del sodio primario nella piscina più interna (sodio caldo). I componenti principali sono il guscio esterno, il deflettore, la piastra porosa (collocata alla base dell'*ACS* che permette il passaggio del sodio) e i tubi di protezione per guidare le barre di controllo. L'*ACS* si estende su una fila di riflettori in acciaio ed è immerso nel sodio ad una profondità di circa 6 m. Lo strato al di sopra del sodio comprende 0,6 m in altezza di argon e può contenere vapore di sodio e aerosol. La maggior parte della struttura è immersa nel flusso di sodio che emerge dal

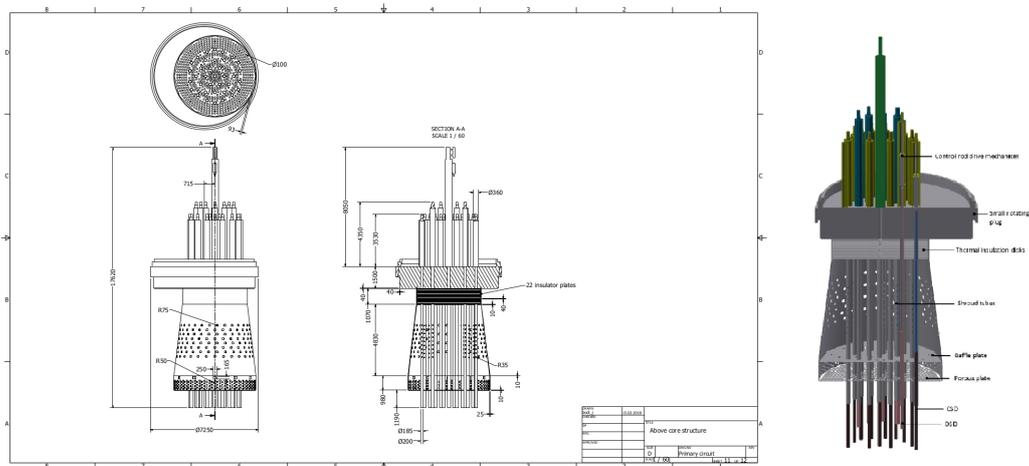


Figura 3.3: Schema e modello dell'ACS con relative dimensioni in scala 1:60

core a temperature che vanno da circa $560^{\circ}C$ al centro dell'uscita del core a $545^{\circ}C$ al pelo libero. La struttura deve essere abbastanza rigida da resistere all'impatto del flusso proveniente dalla parte bassa del core e ad eventi sismici; allo stesso tempo deve risultare flessibile per le forti sollecitazioni di tipo termico.

3.1.3 IHX: Intermediate Heat Exchanger

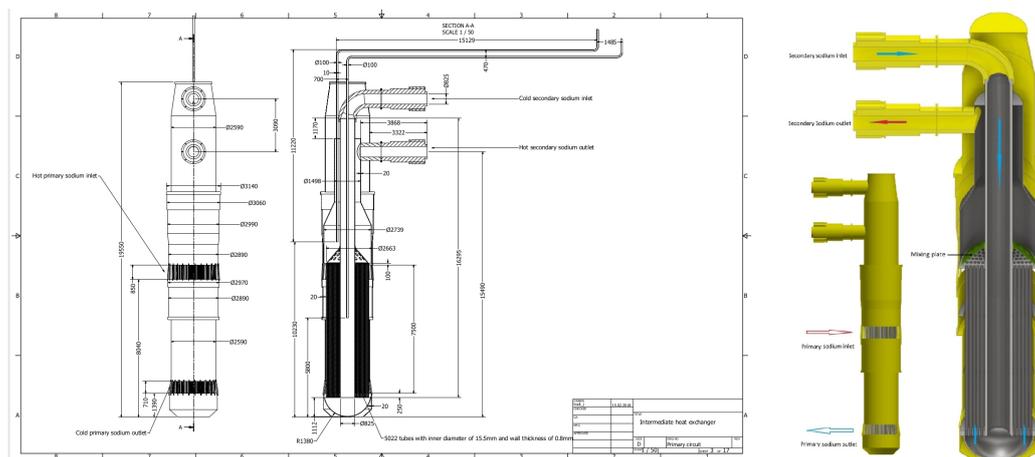


Figura 3.4: Schema e modello dell'IHX con relative dimensioni in scala 1:50

L'IHX è uno scambiatore controcorrente a tubi lineari (si veda figura 3.4), con il sodio secondario che scorre verso il basso attraverso un condotto cen-

trale prima di ruotare verso l'alto nella testata inferiore e fluire verticalmente all'interno dei tubi dello scambiatore di calore, dove viene riscaldato dal sodio primario che scorre verso in basso.

L'unità degli *IHX* è collegata fisicamente e funzionalmente ai circuiti del sodio primario e secondario e come tale deve essere progettata per resistere alla pressione massima di progetto del circuito secondario (circa 55 bar).

L'entrata dalla parte del sodio primario è dotata di una valvola che permette di chiudere l'ingresso ed isolare il circuito primario da quello secondario e dal generatore di vapore, riuscendo, in caso di malfunzionamenti, a lavorare con alcuni *IHX* isolati. Inoltre è saldamente collegato al tetto per ridurre al minimo le vibrazioni ed eventuali perdite di sodio.

3.1.4 Pompa Primaria

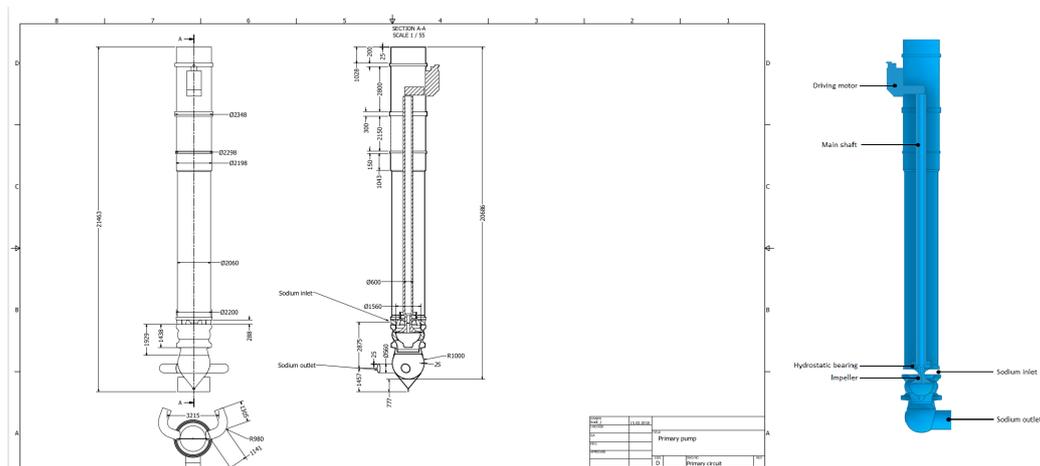


Figura 3.5: Schema e modello di una delle Primary Pump con relative dimensioni in scala 1:55

Le pompe di sodio primarie sono 3 e hanno layout coerente con il design del reattore, alternandosi ogni due *IHX*, così da renderlo compatto. Tra le caratteristiche principali vi sono una girante singola a flusso misto con ingresso del fluido dalla parte superiore e un motore sincrono per permettere alla pompa di operare su tutta la gamma di velocità richieste. La pompa primaria (figura 3.5) comprende un corpo cilindrico ad albero verticale montato e saldato sul tetto del reattore per evitare qualsiasi evento accidentale. Per consentire uno spostamento radiale tra la parte superiore che rimane fredda e la parte inferiore alla temperatura del sodio, la parte inferiore della pompa può piegarsi e adattarsi all'espansione termica. E' inoltre fornita di un diodo

anti flusso-inverso, che consente di limitare il flusso di ritorno attraverso la pompa primaria in caso di guasto. Le perdite di carico attraverso il diodo durante il normale funzionamento sono bassissime.

3.1.5 Sistema Secondario

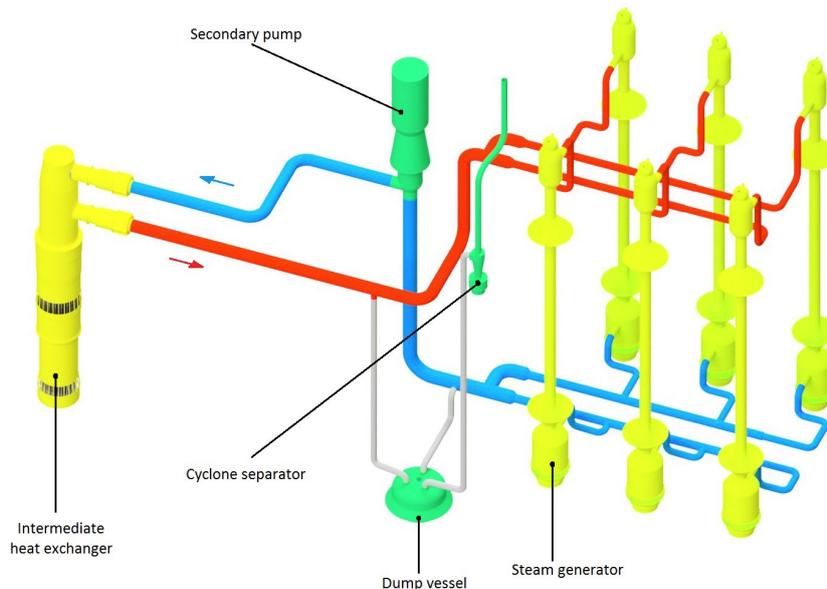


Figura 3.6: Modello di tutto il circuito secondario con ogni componente

Brevemente, il sistema secondario (figura 3.6) trasferisce il calore dagli scambiatori di calore intermedi alle unità del generatore di vapore durante la produzione di potenza e durante lo smaltimento del calore di residuo (attraverso il sistema terziario *acqua/vapore* dopo l'arresto del reattore). Può essere visto come un loop non-radioattivo esterno al reattore, visto che il sodio secondario e l'acqua non entrano in contatto con le barre di combustibile. Ogni loop secondario comprende un *IHX*, una Pompa secondaria, una Unità generatore di vapore con 6 moduli per ogni unità e un sistema di smaltimento del calore residuo(DHRS-1) La portata di sodio in ciascun circuito è di $10800 \frac{m^3}{h}$ ($2541 \frac{kg}{s}$) e la potenza termica di ciascun generatore di vapore è di $100 MW_{th}$.

3.2 Redan

Il vessel interno è una struttura di fabbricato asimmetrico che comprende una parte conica (*REDAN*), un lembo cilindrico superiore all'interno del vessel

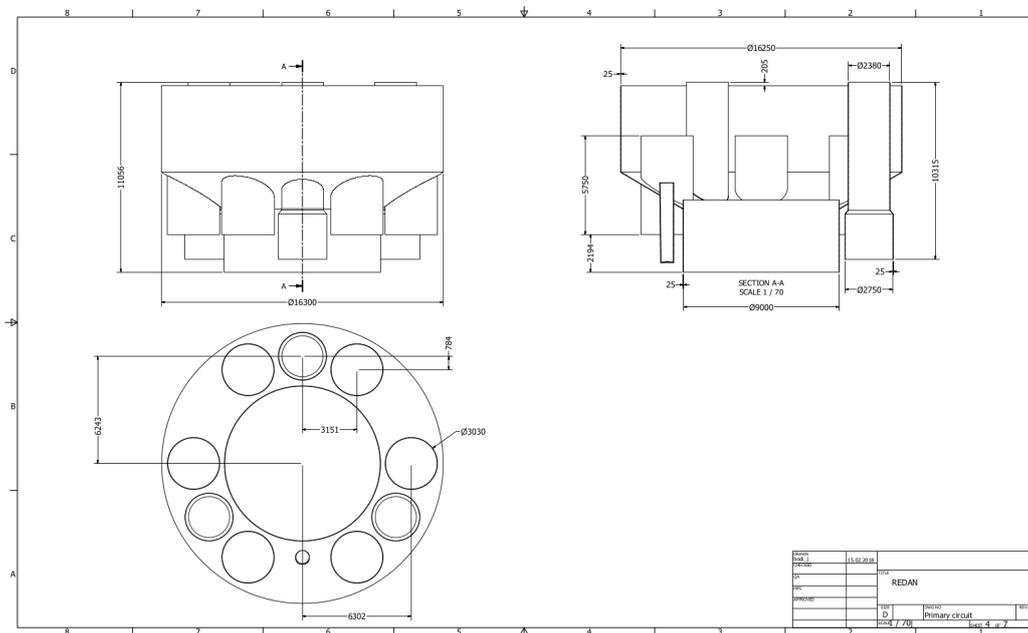


Figura 3.7: Schema del REDAN in scala 1:70

e un minimo di 9 penetrazioni nel *REDAN:6* per l'*IHX* e 3 per le pompe primarie. Il vessel interno separa la piscina calda, che contiene i sottosistemi del core e gli ingressi dell'*IHX*, dalla piscina fredda, in cui sono situati le uscite dell'*IHX* e gli ingressi delle pompe primarie, fornendo una barriera a tenuta stagna [8].

Il modello conico del *REDAN* viene preferito a quello ogivale precedente, per semplificare la produzione e renderlo meno sensibile alle deformazioni plastiche, riducendo anche la superficie soggetta alla differenza di pressione tra i collettori.

Capitolo 4

Creazione del mesh per la simulazione termo-fluidodinamica del REDAN

La creazione di una griglia computazionalmente valida del *REDAN*, si veda figura 3.7, è l'oggetto di questa tesi. La sua realizzazione è stato possibile dividendo il dominio in maniera coerente e facendo particolare attenzione ai componenti che ne fanno parte.

4.1 Analisi della mesh

Analizzare la qualità di una mesh significa visualizzare, nel complesso, il modo in cui le grandezze geometriche caratterizzanti gli elementi di quest'ultima si rapportano tra loro. Dopo aver definito le geometrie principali degli elementi che compongono le mesh (triangoli, quadrilateri ...) si passerà alla definizione di alcuni parametri che ne indicano la qualità [10].

Triangoli. I triangoli rappresentano la geometria più semplice, ma anche la più presente, di una mesh. Un triangolo presenta 3 lati e 3 angoli che possono differire in lunghezza/ampiezza. Nel caso fossero tutti uguali, come mostrato in figura 4.1, verrà chiamato *equilatero* con lati

$$\mathbf{L}_0 = \mathbf{P}_2 - \mathbf{P}_1, \mathbf{L}_1 = \mathbf{P}_2 - \mathbf{P}_0, \mathbf{L}_2 = \mathbf{P}_0 - \mathbf{P}_1;$$

di lunghezza

$$L_0 = \|\mathbf{L}_0\|, L_1 = \|\mathbf{L}_1\|, L_2 = \|\mathbf{L}_2\|$$

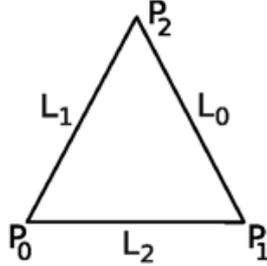


Figura 4.1: Numerazione dei vertici e dei lati di un elemento triangolare

ed area

$$A = \frac{1}{2} \|\mathbf{L}_0 \wedge \mathbf{L}_1\| = \frac{1}{2} \|\mathbf{L}_1 \wedge \mathbf{L}_2\| = \frac{1}{2} \|\mathbf{L}_2 \wedge \mathbf{L}_0\|.$$

Inoltre è possibile definire il raggio della circonferenza inscritta come

$$r = \frac{2A}{\|\mathbf{L}_0\| + \|\mathbf{L}_1\| + \|\mathbf{L}_2\|}$$

ed il raggio della circonferenza circoscritta:

$$R = \frac{\|\mathbf{L}_0\| \|\mathbf{L}_1\| \|\mathbf{L}_2\|}{2r(\|\mathbf{L}_0\| + \|\mathbf{L}_1\| + \|\mathbf{L}_2\|)}.$$

Uno dei parametri per identificare la qualità dell'elemento di una mesh triangolare è l'*triangle edge ratio* e il *triangle radius ratio*. Indicando con q il valore della qualità, il primo è definito come

$$q = \frac{L_{max}}{L_{min}},$$

mentre il secondo

$$q = \frac{R}{2r}.$$

dove L_{max} ed L_{min} sono rispettivamente il lato più lungo e più corto del triangolo in esame. Valori ottimali sono in un range che va da 1 a 1,3.

Quadrilateri. Per quadrilateri si intendono poligoni di 4 lati che, come mostrato in figura 4.2, possono essere più o meno regolari. I lati di un quadrilatero sono definiti come

$$\mathbf{L}_0 = \mathbf{P}_1 - \mathbf{P}_0, \mathbf{L}_1 = \mathbf{P}_2 - \mathbf{P}_1, \mathbf{L}_2 = \mathbf{P}_3 - \mathbf{P}_2, \mathbf{L}_3 = \mathbf{P}_0 - \mathbf{P}_3$$

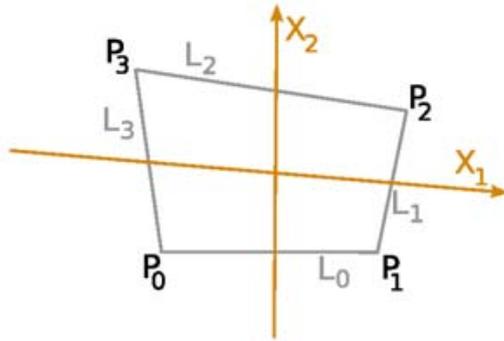


Figura 4.2: Numerazione dei vertici e dei lati di un quadrilatero

e diagonali

$$D_0 = P_2 - P_0, D_1 = P_3 - P_1,$$

che dividono il quadrilatero in 4 regioni. Se il quadrilatero ha tutti i lati e gli angoli uguali verrà chiamato *quadrato*. Un parametro per analizzare la qualità dell'elemento di mesh è, come in precedenza e con la stessa definizione, l'*edge ratio*, ovvero il rapporto tra il lato più lungo e quello più corto del quadrilatero.

Tetraedri. Il tetraedro è una figura tridimensionale che ha 4 facce tutte triangolari. E' l'elemento più presente all'interno di mesh 3-D e per questo lo studio della qualità risulta molto importante. I lati di un tetraedro sono definiti come (si veda la figura 4.3)

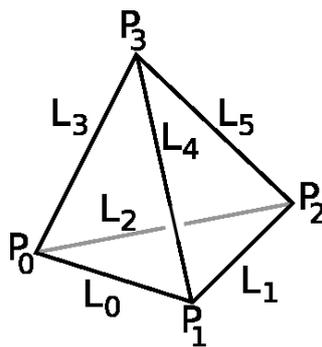


Figura 4.3: Numerazione dei vertici e dei lati di un tetraedro

$$L_0 = P_1 - P_0, L_1 = P_2 - P_1, L_2 = P_0 - P_2,$$

$$\mathbf{L}_3 = \mathbf{P}_3 - \mathbf{P}_0, \mathbf{L}_4 = \mathbf{P}_3 - \mathbf{P}_1, \mathbf{L}_5 = \mathbf{P}_3 - \mathbf{P}_2;$$

con area

$$\frac{1}{2}(\|\mathbf{L}_2 \wedge \mathbf{L}_0\| + \|\mathbf{L}_3 \wedge \mathbf{L}_0\| + \|\mathbf{L}_4 \wedge \mathbf{L}_1\| + \|\mathbf{L}_3 \wedge \mathbf{L}_2\|),$$

e volume

$$V = \frac{(\mathbf{L}_2 \wedge \mathbf{L}_0) \cdot \mathbf{L}_3}{6}.$$

Come per i triangoli, anche qui è possibile definire il raggio della circonferenza inscritta

$$r = \frac{3V}{A},$$

e circoscritta

$$R = \frac{\|\mathbf{L}_3\|^2(\mathbf{L}_2 \wedge \mathbf{L}_0) + \|\mathbf{L}_2\|^2(\mathbf{L}_3 \wedge \mathbf{L}_0) + \|\mathbf{L}_0\|^2(\mathbf{L}_3 \wedge \mathbf{L}_2)}{12V}.$$

Anche in questo caso il parametro per verificare la qualità di un elemento tetraedrico è il *radius ratio* ma è definito diversamente, ovvero

$$q = \frac{R}{3r}$$

sempre compreso tra 1 e 1,3.

Piramidi. Le piramidi sono costituite da 4 facce triangolari poggianti su una base quadrangolare (si veda la figura 4.4). Essendo che da una mesh bi-

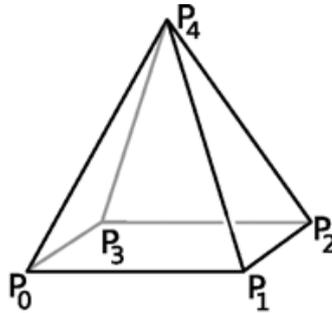


Figura 4.4: Numerazione dei vertici di una piramide

dimensionale strutturata si genereranno piramidi passando al corrispondente

3-D, sono molto importanti da analizzare e sono la struttura sulle quali si verranno a formare i tetraedri che riempiranno il dominio (ogni faccia della piramide costituisce la base per il tetraedro). Il parametro utilizzato per studiare la qualità è il *pyramid Volume*

$$q = V$$

dove V è il volume della piramide ed è calcolato dividendo la piramide in due tetraedri opportunamente orientati. Il numero 0 è simbolo di una buona qualità dell'elemento di mesh (la somma di volumi orientati è possibile che dia 0 come risultato se i tetraedri sono perfettamente simmetrici).

4.2 Divisione del dominio e creazione della geometria

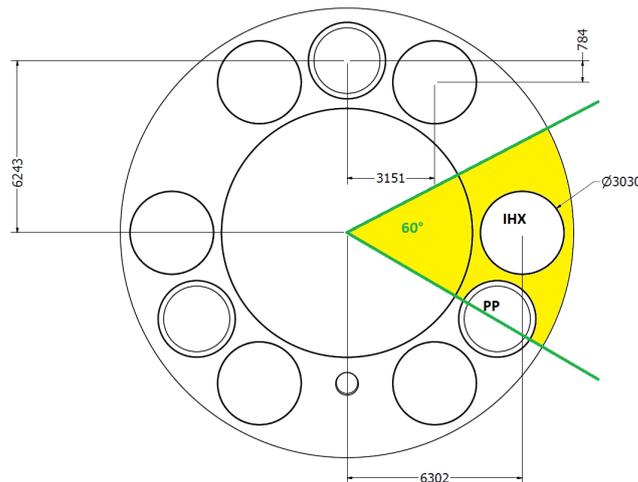


Figura 4.5: Divisione del dominio in un angolo pari a $\frac{\pi}{3}$

Analizzando la geometria del *REDAN*, si può notare una forte simmetria tra i componenti ovvero 3 gruppi formati da 2 *IHX* con in mezzo una pompa primaria e l'*ACS* che ricopre tutto l'angolo di $\frac{\pi}{3}$. Colorata in giallo nella figura 4.5 è rappresentata la zona di interesse. Abbiamo scelto di dividere il *REDAN* in angoli pari a $\frac{\pi}{3}$ includendo tutto l'esterno dell'*IHX* e metà della pompa primaria. A completare il profilo c'è uno spicchio di *ACS* con il relativo ingresso del sodio dal basso.

Il primo passo compiuto è stato individuare una geometria base da cui partire, all'inizio concepita senza considerare *PP* e *IHX* come mostrato in figura 4.6.

Il perimetro principale comprende un primo tratto di *ACS* fino al livello del sodio caldo nella piscina primaria, il tratto orizzontale-verticale della parete esterna del *REDAN* e il tratto diagonale che attraversa preliminarmente l'*IHX* e la *PP*. Una volta definito il profilo e la sua area, è stato creato il volume e le altre superfici fisiche attraverso una rivoluzione intorno ad un'asse passante per il centro del *REDAN* pari a $\frac{\pi}{3}$. In questo modo abbiamo ottenuto le superfici principali di cui avevamo bisogno, ovvero l'ingresso del sodio dal basso della piastra porosa, la porzione di *ACS*, la parete esterna del *REDAN* e uno spicchio superiore. Successivamente sono state disegnate le due circonferenze base per l'*IHX* e la *PP*.

Entrambi i profili sono stati ottenuti poi per estrusione di un cilindro fino alla quota rilevata sul cad, ma in maniera leggermente diversa: la *PP* è rappresentata da un unico cilindro che scava interamente il profilo mentre l'*IHX* è una sovrapposizione di tre cilindri diversi per "funzione" poiché quello centrale rappresenta l'ingresso del sodio caldo e i due laterali fungono da parete. Per il risultato finale della geometria si veda la figura 4.7.

4.2.1 Superfici Fisiche

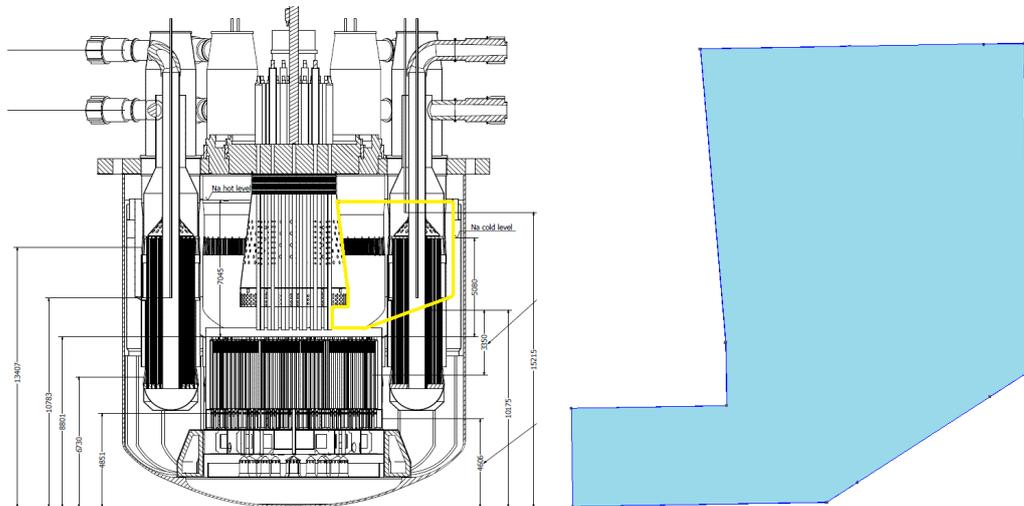


Figura 4.6: Perimetro principale del modello da realizzare. Si noti come la linea gialla diagonale taglia sia l'*IHX* e la *PP*. Affianco, in azzurro, il primo risultato.

Le superfici fisiche sono tutte quelle che compongono il profilo e hanno un preciso scopo in una simulazione fluidodinamica. Possono essere sia singole

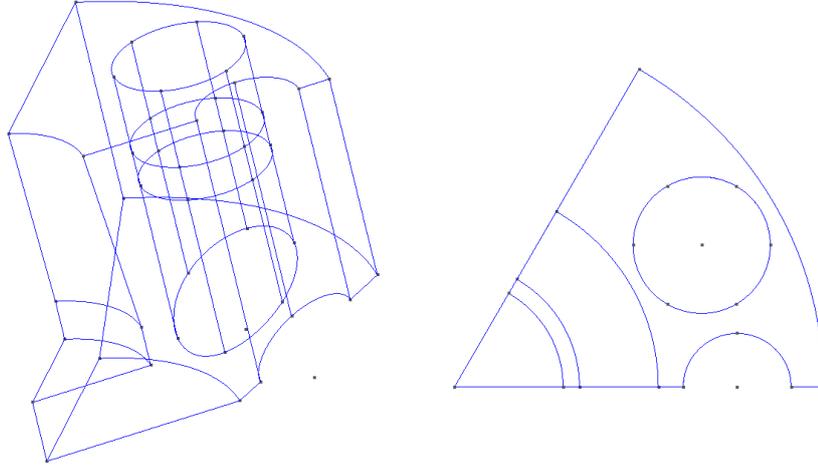


Figura 4.7: Visione globale e dall'alto del modello finale. Nella figura di sinistra si possono vedere i 3 cilindri dell'*IHX*, la semicirconferenza della *PP* e le pareti del *REDAN*

superficie	specifiche
HotSodiumInlet	ingresso del sodio primario all'interno
sodiumCoreInlet	ingresso del sodio attraverso il core
REDAN	pareti esterne di tutta la rotazione pari a $\frac{\pi}{3}$
primaryPump	parete esterna della <i>PP</i>
aboveCoreStructures	parete della porzione di ACS
underIHXSodium	pareti della piscina interna
upperIHX e lowerIHX	cilindro superiore e inferiore dello scambiatore
HotNA	strato superiore di sodio caldo

Tabella 4.1: Regioni dello spazio considerate nel modello

che raggruppamenti di più superfici geometriche, ma accomunate dalla stessa funzione nella simulazione fluidodinamica. Nello sviluppo della mesh sono state utilizzate varie tipologie di superfici sulle quali sono poi state imposte le condizioni al contorno che chiameremo wall, inlet ed outlet.

Con *wall* indichiamo le pareti che non possono essere attraversate dal fluido, e in genere corrispondono ad una condizione *no slip* (scorrimento nullo) per la velocità. Su queste superficie è possibile imporre una temperatura fissata o un flusso di calore. Il caso più comune è quello di flusso di calore nullo, che

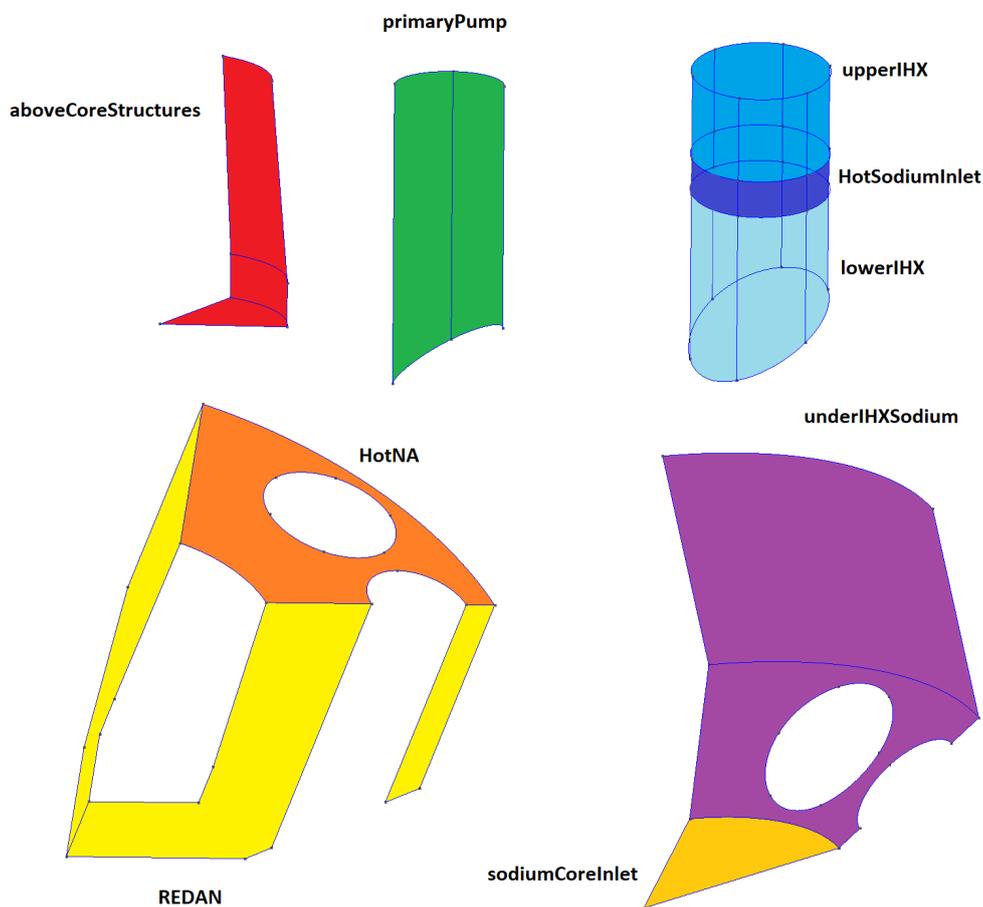


Figura 4.8: Visualizzazione delle superfici fisiche con relativo nome. Colori diversi indicano superfici fisiche diverse.

corrisponde ad una superficie adiabatica.

Con *inlet* indichiamo una porzione di bordo da cui il fluido entra nel dominio computazionale. In generale, per questa tipologia di superficie la velocità normale alla faccia viene imposta, assieme ad alcune quantità che fissano il livello di turbolenza desiderato. Per la temperatura è necessario imporre un valore prefissato.

Con *outlet* designiamo una superficie da cui il fluido lascia il dominio computazionale. In questo caso si impone il valore della pressione che si desidera avere su questa faccia (in generale 0, in modo che tutti i valori di pressio-

ne all'interno del dominio si possano considerare come differenziali rispetto all'uscita). In questo caso è possibile imporre una temperatura prefissata, o considerare la superficie adiabatica.

Per soddisfare il vincolo di incomprimibilità, è necessario che, se il dominio prevede una superficie di *inlet*, sia presente almeno una superficie di *outlet*. Per il caso del modello del REDAN consideriamo diverse regioni di spazio come indicate in Tabella 4.1 che possono essere visualizzate in figura 4.8 in diversi colori.

4.3 Creazione della Mesh

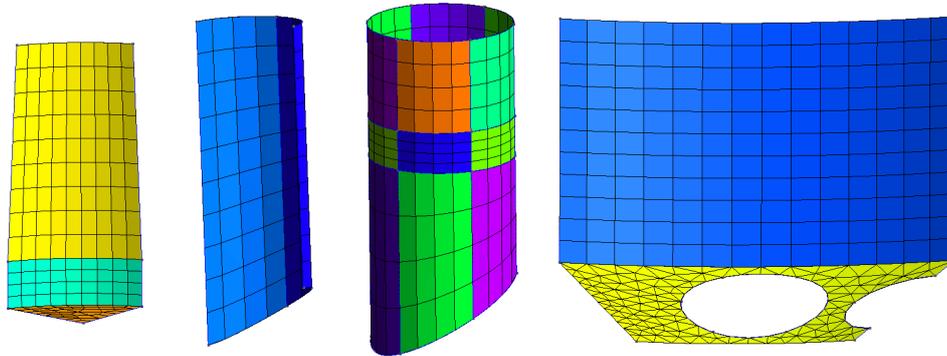


Figura 4.9: Visualizzazione delle mesh strutturate. Da sinistra a destra: ACS, PP, IHX, underIHXSodium

Durante l'attività sono state realizzate diverse mesh per le simulazioni, ognuna diversa per caratteristiche (numero di triangoli, tetraedri, ...) e per algoritmo di generazione. Infatti *Gmsh* mette a disposizione dell'utente diversi programmi per la generazione delle mesh, ovvero:

mesh 2-D : Automatic, Mesh Adapt, Delaunay, Packing of Parallelograms

mesh 3-D : Delaunay, Frontal, HXT

La struttura della mesh è mista, ovvero dove possibile la mesh è stata strutturata, in modo da seguire la geometria del profilo ed avere dei quadrilateri per lo più regolari, mentre dove non si può sono stati lasciati dei triangoli. Ogni mesh risulta diversa anche per la risposta alle simulazioni che fornisce:

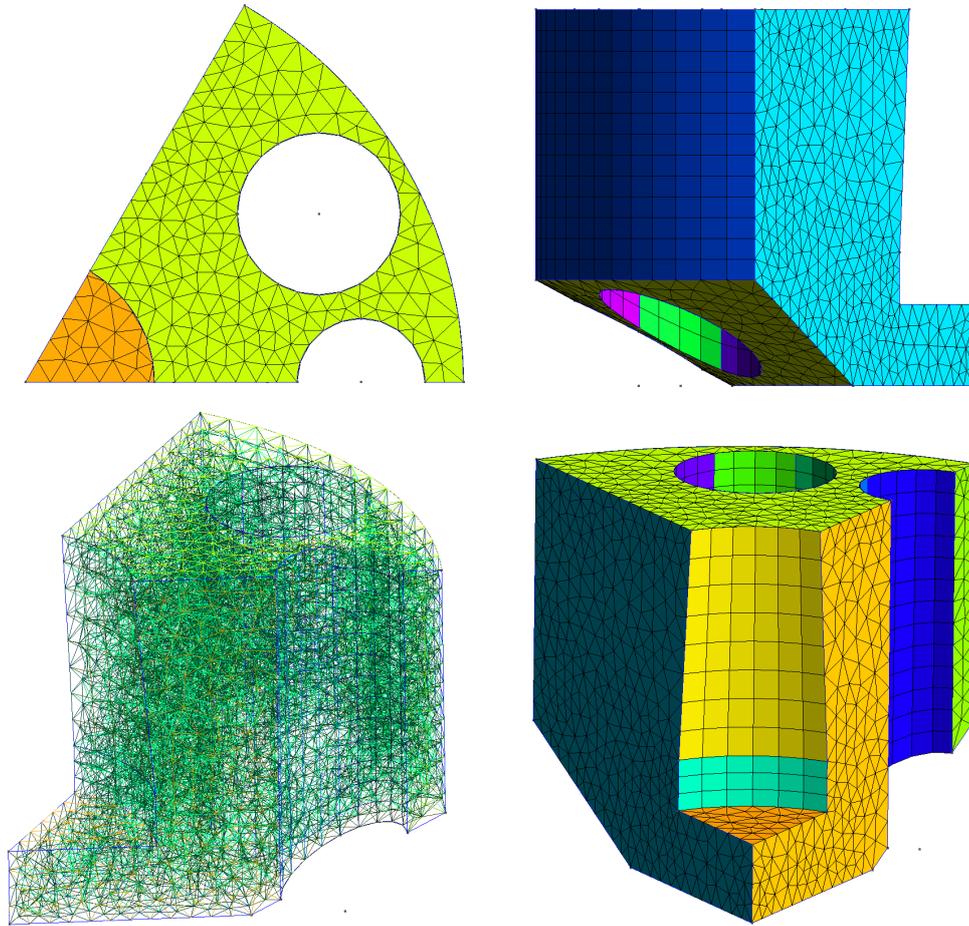


Figura 4.10: In senso orario: Mesh 2-D vista dall'alto, lateralmente e frontalmente, Mesh 3-D. (mista)

i risultati di alcune simulazioni sembrano essere coerenti con quelli attesi, mentre in altri casi la mesh porta a soluzioni non fisiche.

Il processo di generazione di una mesh strutturata è simile per tutti gli algoritmi sopracitati e verrà racchiusa nella sezione successiva. La mesh finale mista verrà presentata nell'apposita sezione.

E' stato possibile strutturare la mesh lì dove la geometria presentava una simmetria e un perimetro di 4 lati. Infatti in *PyGmsh* attraverso i comandi `transfinite` e `recombine` è possibile trasformare mesh triangolari in quadrangolari seguendo la geometria desiderata.

Il procedimento è il seguente: per prima cosa si applica il `transfinite` alle 4 linee che formano il perimetro, specificando il numero di punti desiderati per ogni linea. Successivamente si applica il `transfinite` alla superficie che le

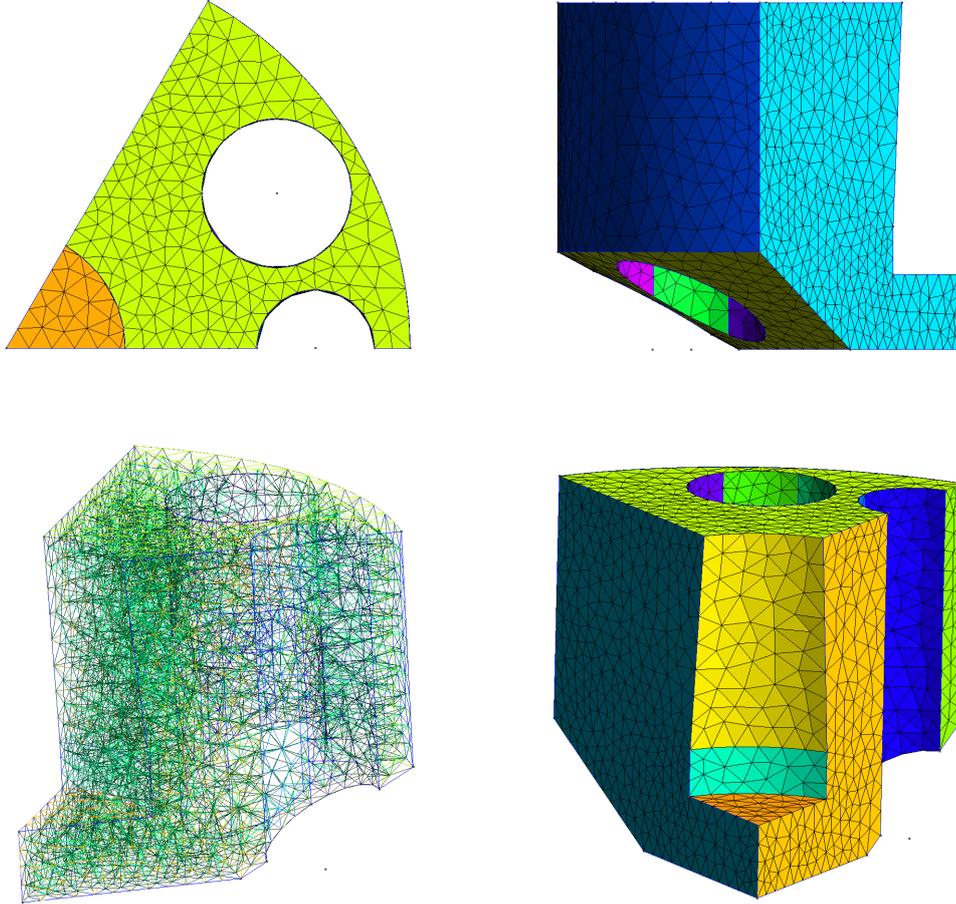


Figura 4.11: In senso orario: Mesh 2-D vista dall'alto, lateralmente e frontalmente, Mesh 3-D. (non strutturata)

zona	algoritmo	raffinamento
ACS	transfinite 7 4 e 9	progression 1
PP	transfinite 14 e 6	progression 1
upperIHX	transfinite 5 e 7	progression 1
HotSodiumInlet	transfinite 5	progression 1
lowerIHX	transfinite 5 e 9	progression 1
sodiumUnderIHX	transfinite 15 e 14	progression 1
underIHXSodium	transfinite 12 e 15	progression 1

Tabella 4.2: Divisione in mesh delle zone geometriche

4 linee individuano (questa procedura funziona solo se le due coppie di linee affacciate hanno lo stesso numero di punti). Una volta fatto ciò si applica il **recombine** a tale superficie così da convertire le coppie di triangoli in un singolo quadrilatero ed infine generare la mesh.

Le uniche superfici della nostra geometria sulle quali è stato possibile applicare questo approccio sono: i 3 cilindri dell'*IHX*, la *PP*, una parte dell'*underIHX Sodium* e una parte di *ACS* come mostrato in figura 4.9.

Nel dettaglio la suddivisione è stata fatta come in Tabella 4.2 dove con *progression* è indicato il rapporto tra la lunghezza di una cella e la successiva (in questo caso pari ad 1 perché sono delle stesse dimensioni).

Tutte le altre superfici fisiche, e una parte di quelle precedenti, non presentando 4 lati né assi di simmetria, avranno una mesh a triangoli o quadrilateri irregolari. Una volta creata la mesh 2-D, è necessario definire la mesh tridimensionale, ovvero generare una fitta rete di celle all'interno del dominio. Il risultato (mesh mista 2-D e mesh 3-D) è mostrato in figura 4.10, mentre in figura 4.11 è mostrato lo stesso modello con mesh completamente non strutturata. Le due serviranno per confrontare i risultati di eventuali simulazioni fluidodinamiche.

Capitolo 5

Analisi dei risultati

In questo capitolo verrà simulato l'ingresso del sodio attraverso il core fino all'uscita nell'*IHX* sulle mesh da noi realizzate. Per farlo si utilizzerà *OpenFOAM*, sotto opportune condizioni che verranno poi specificate, e successivamente i risultati analizzati con *Paraview*. Inoltre si mostreranno, dopo una breve introduzione, la diversa qualità delle mesh attraverso dei precisi parametri.

5.1 Tipologia di problema e condizioni iniziali

superficie	condizioni al contorno	specifiche
REDAN	<i>symmetry</i>	
HotSodiumInlet	<i>inletOutlet</i>	uniform (0 0 2.22e - 4)
sodiumCoreInlet	<i>fixedValue</i>	uniform (0 0 2.22e - 4)
Altre	<i>noSlip</i>	

Tabella 5.1: Condizioni al contorno per la velocità

OpenFOAM mette a disposizione diversi strumenti per la risoluzione delle equazioni e della turbolenza. Per il nostro caso, è stato scelto *pimpleFoam*, risolutore non stazionario per fluidi incomprimibili su larghi timestep e combinazione degli algoritmi PISO (Pressure Implicit with Splitting of Operator) e SIMPLE (Semi-Implicit Method for Pressure-Linked Equations). La simulazione è puramente fluidodinamica, perciò non verranno analizzate le variazioni di Temperatura e gli scambi di calore. Il primo passo da fare è fissare la condizione iniziale per la velocità all'interno del dominio, nella cartella $0/U$, attraverso il comando *internalField uniform(0 0 0)*. Questo permette di impostare inizialmente a 0 il valore della velocità, in tutte e 3 le direzioni, in

superficie	condizioni al contorno	specifiche
REDAN	<i>symmetry</i>	
HotSodiumInlet	<i>fixedValue</i>	uniform(0)
Altre	<i>zeroGradient</i>	

Tabella 5.2: Condizioni al contorno per la pressione

ogni cella del dominio. Successivamente devono essere settate le condizioni al contorno per ogni superficie fisica e nel caso della velocità le troviamo nella tabella 5.1 dove *symmetry* indica il comportamento speculare delle due facce del *REDAN*, *noSlip* indica che ad un bordo solido la velocità relativa è nulla e le altre due sono condizioni per l'ingresso/uscita del sodio.

Per la pressione non è necessario impostare la condizione iniziale, visto che nell'equazione della pressione non compare esplicitamente la dipendenza temporale. Perciò le condizioni al contorno sono in Tabella 5.2 dove *zeroGradient* pone a 0 il valore del gradiente del campo considerato su una determinata superficie fisica. Inoltre è bene precisare che la pressione, in *pimpleFoam* non è quella statica p_s ma cinematica p_k tale che:

$$p_k = \frac{p_s}{\rho}$$

dove ρ è la densità del fluido considerato (costante nel caso di fluido incomprimibile).

Gli altri parametri da controllare sono:

k : Energia cinetica turbolenta

ϵ : rate di dissipazione turbolenta

ν_t : viscosità turbolenta

oltre a quelli che rimangono costanti durante il moto come g , l'accelerazione di gravità, e ν ovvero la viscosità cinematica in questo caso pari a $10^{-7} \frac{m^2}{s}$. Il valore della velocità all'ingresso, u_{inl} , è stato trovato una volta fissato il numero di *Reynolds* $Re = 20000$ [9]. Infatti dalla (1.34), si ottiene:

$$u_{inl} = \frac{Re \cdot \nu}{L} = \frac{20000 \cdot 10^{-7}}{9} = 2.22 * 10^{-4} \frac{m}{s}$$

dove L è il diametro idraulico (coincidente col diametro del REDAN essendo l'inlet circolare). la durata della simulazione è specificato nel file `controlDict` ed è pari a 100 secondi, dt fisso a 0.5 con scrittura dei risultati ogni 5. L'estensione temporale è stata scelta in base alla convergenza delle grandezze sopracitate.

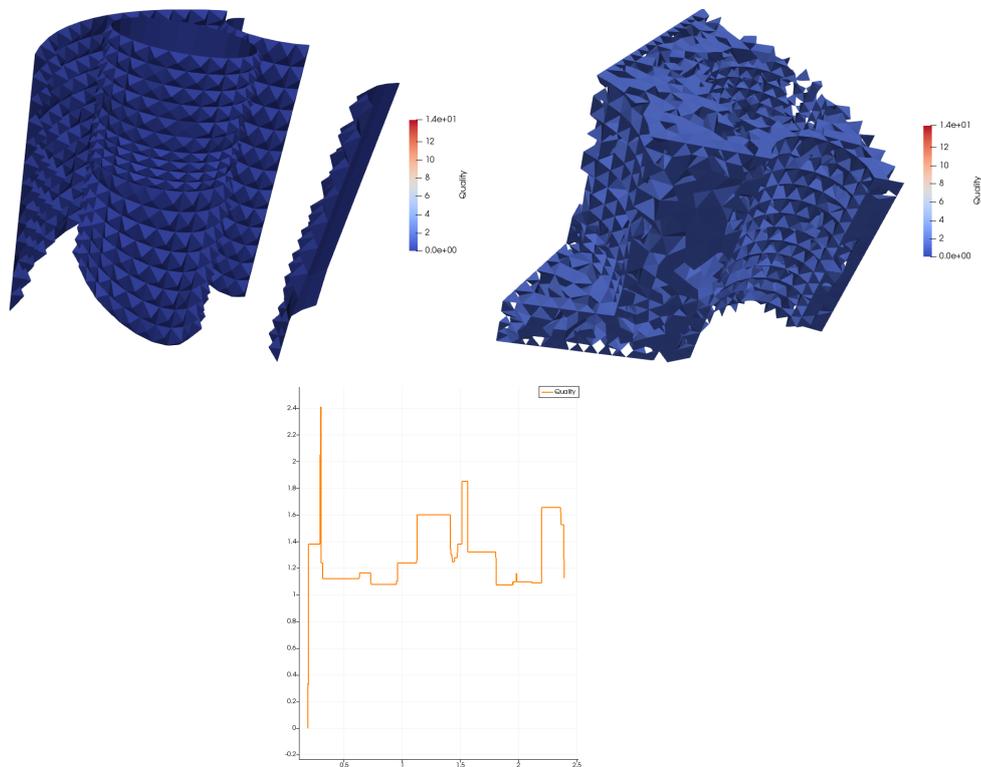


Figura 5.1: In alto: elementi costituenti la mesh 3-D mista su due scale di visualizzazione. In basso: plotting della mesh quality

5.2 Confronto tra le mesh e campi di soluzione

Attraverso gli strumenti offerti da *Paraview* si analizzeranno i diversi risultati provenienti dalle simulazioni, confrontandoli in funzione della mesh che li ha prodotti.

È possibile analizzare il valore di q definito in precedenza attraverso il filtro *meshquality* di *Paraview* e impostando, attraverso il comando *threshold*, una scala di visualizzazione. Sono stati trattati separatamente i casi di mesh mista e non strutturata.

Nella figura 5.1, si possono notare diverse scale di mesh quality. La prima scala va da un valore di $q=0$ fino a 0.99. Come conseguenza sono rappresentate soltanto le piramidi che hanno come base i quadrilateri delle superfici con mesh strutturata, dove q assume un valore pari a 0 o prossimo ad esso. Ciò accade perché i quadrilateri sono più o meno tutti regolari e i 2 tetraedri in cui viene scomposta la piramide hanno volume pressoché identico. La seconda scala invece va da un valore di $q=1$ fino ad 1.3 e si può notare che,

unendo le celle di quest'ultime due scale, si riesca a riempire quasi tutto il dominio, indice di una mesh di buona qualità.

In basso invece viene graficato l'andamento della *mesh quality* su una linea che parte dalla parete dell'*ACS* fino all'ingresso dell'*IHX*. Essendo il primo punto della linea sulla parete dove la mesh è strutturata, la qualità è all'inizio 0 per poi aumentare abbastanza velocemente fino a 2.4. Questo è una conseguenza della mesh mista, dove nella generazione del 3-D si incontrano poligoni regolari con irregolari impattando sulla qualità.

In figura 5.2 invece viene presentata la stessa analisi riferendosi alla mesh non strutturata. È stata ovviamente omessa la scala da 0 a 0.99 visto che non esistono quadrati su cui costruire piramidi e q parte da un valore di 1 fino ad 1.3 e successivamente da 1.3 fino a 2.3. Anche in questo caso la

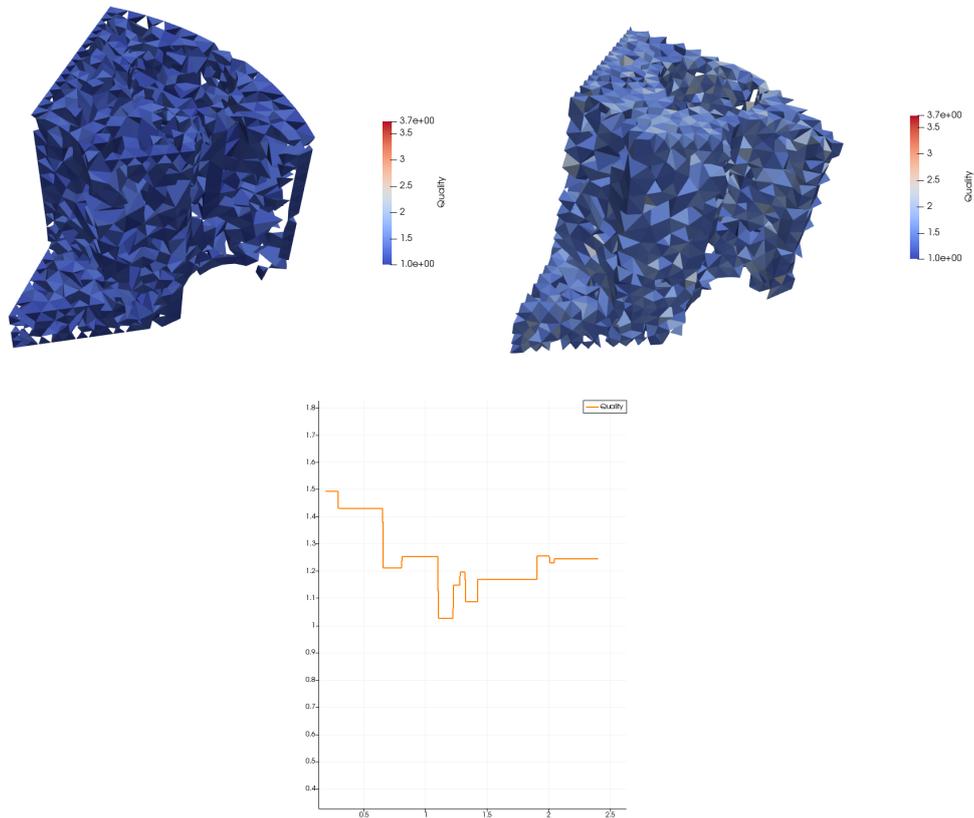


Figura 5.2: In alto: elementi costituenti la mesh 3-D non strutturata su due scale di visualizzazione. In basso: plotting della mesh quality

mesh è di buona qualità e rispetto al caso precedente e nel grafico varia meno bruscamente con picchi di $q=1.5$.

Le *streamlines* sono una famiglia di curve istantaneamente tangenti al vettore velocità del flusso e mostrano la direzione in cui a qualsiasi istante temporale si muoverà l'elemento di fluido. Streamlines differenti dello stesso flusso non possono intersecarsi visto che il fluido è incomprimibile. Per visualizzarle è stato creato un piano con il comando `plane` passante per i punti di coordinate $(0\ 0\ 9)(4\ 0\ 9)(0\ 4\ 9)$, in modo da intersecare l'ingresso del sodio dal *core*. Come mostrato in figura 5.3, si possono notare alcune differenze. Per la mesh mista, le streamlines partono dal *sodiumCoreInlet*, si aprono intorno al *lowerIHX* ed escono dall'*HotSodiumInlet* in maniera abbastanza uniforme. Per la mesh non strutturata invece, le streamlines si raggruppano sulla

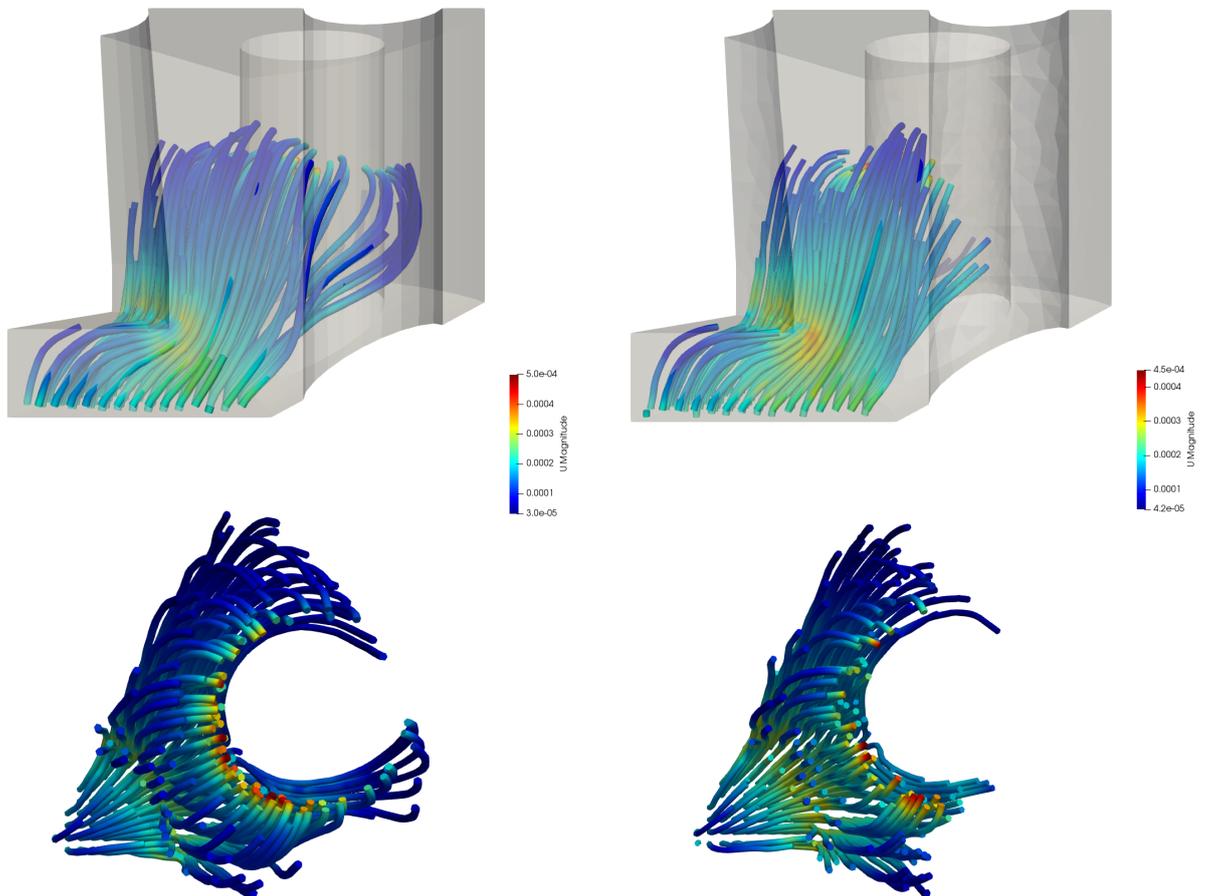


Figura 5.3: Streamlines della velocità con relativa legenda. Nella colonna di sinistra: mesh mista; in quella di destra: mesh non strutturata

parte frontale dell'uscita. È interessante analizzare come varia la velocità di un elemento di fluido dal punto più lontano a quello più vicino all'uscita nell'*IHX*. Abbiamo ottenuto, attraverso il comando `slice` di paraview, una sezione dell'uscita e realizzato su di essa una linea che parte dall'*ACS* fino all'*HotSodiumInlet*. Successivamente, attraverso il comando `plot over line`, sono stati calcolati i valori di velocità lungo tutta la linea, come mostrato in figura 5.4. Dal grafico si può notare come la mesh mista (linea blu) sia più precisa rispetto a quella non strutturata (linea rossa). Infatti la velocità, alla parete dell'*ACS*, nel primo caso è pressoché 0 in accordo con la *noSlip* condition. Nel secondo caso invece è dell'ordine di $10^{-5} \frac{m}{s}$. In entrambi i casi si può notare un'aumento di velocità sull'ingresso frontale dell'*IHX* piuttosto che su quello posteriore.

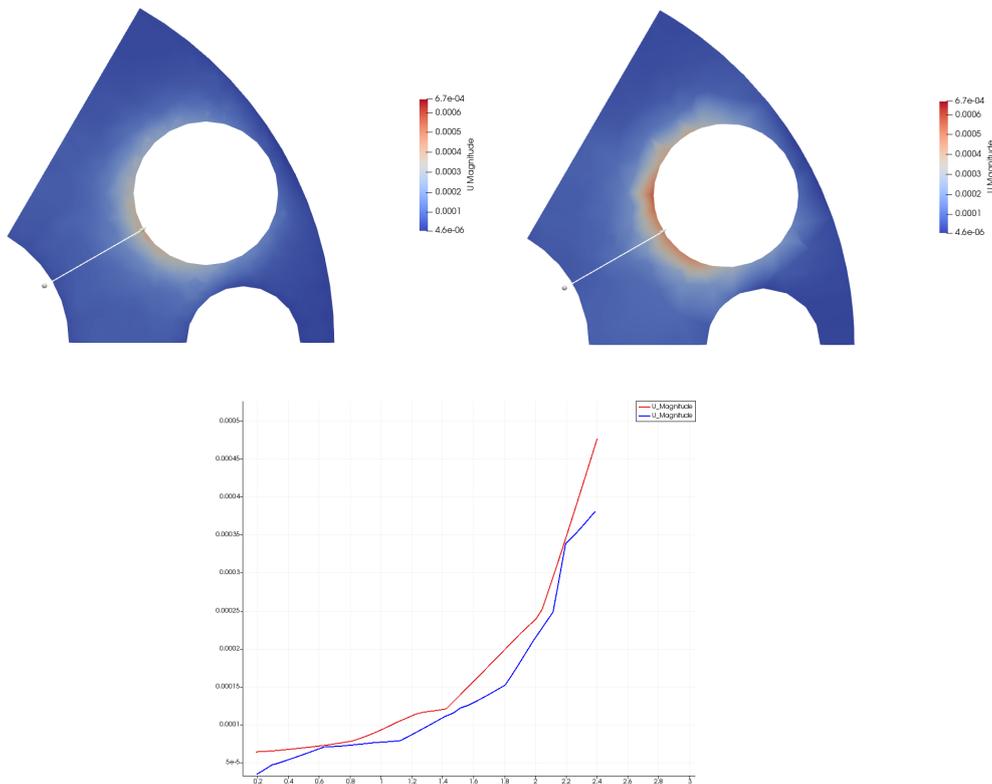


Figura 5.4: Nella colonna di sinistra: clip della mesh mista; in quella di destra: clip mesh non strutturata; in basso: plotting della velocità in funzione della distanza tra *ACS* e *HotSodiumInlet*. La retta blu rappresenta la mesh mista mentre quella rossa la non strutturata

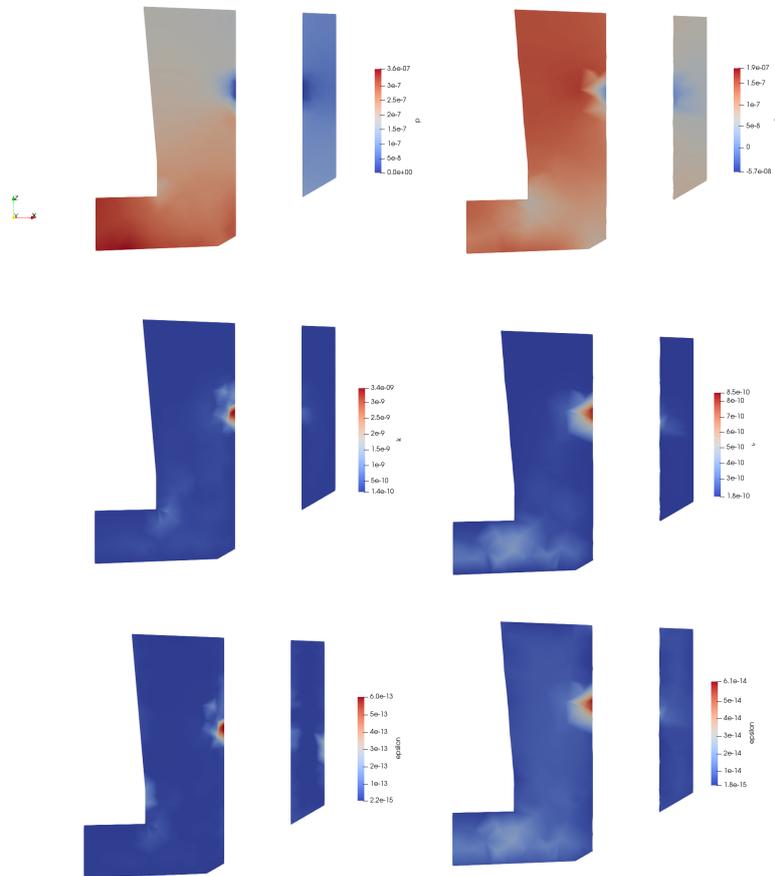


Figura 5.5: Nella colonna di sinistra: clip della mesh mista; nella colonna di destra: clip della mesh non strutturata. Dall'alto verso il basso, parallelamente si possono notare: pressione, energia cinetica turbolenta k e la sua dissipazione ϵ

Sempre attraverso il comando `slice` è stata ottenuta una sezione lungo il raggio dell'*ACS* che taglia il reattore a metà. E' possibile visualizzare, attraverso il colore e la scala in figura 5.5, la pressione, energia cinetica turbolenta k e la sua dissipazione ϵ . Nella mesh mista, il valore della pressione raggiunge lo 0 (relativo) all'ingresso dell'*IHX*, mentre nell'altro caso l'ordine è di 10^{-8} . Dai grafici di energia cinetica turbolenta k e la sua dissipazione ϵ si può notare invece come la dispersione dei valori (macchie chiare) sia molto più marcata nella mesh non strutturata rispetto a quella mista.

Conclusioni

In questo elaborato di tesi è stato realizzato un modello computazionale della piscina calda di un reattore refrigerato al sodio. Dopo aver presentato le equazioni della fluidodinamica ed i programmi utilizzati nel capitolo 1 e 2, nel capitolo 3 è stato descritto il progetto ESFR-SMART, iniziato il 1 settembre 2017 e con realizzazione prefissata entro 48 mesi. Il reattore è composto da strutture molto complesse, tra cui il *REDAN*, che separa le due piscine interna ed esterna ed è quindi un componente termicamente molto sollecitato. Tale componente, racchiudendo una piscina dalla geometria complessa e articolata, mal si presta alla simulazione con codici di sistema a parametri concentrati. Per la sua simulazione risulta molto più indicato un modello CFD e per questo motivo esso è stato scelto come modello per questo elaborato di tesi.

Una parte fondamentale nella simulazione CFD è la realizzazione di una buona mesh, che permetta la risoluzione dei campi di interesse in maniera accurata e col minimo tempo di calcolo necessario. A tale scopo in questa tesi abbiamo sviluppato una strategia ad hoc, descritta in dettaglio nel capitolo 4, basata sulla divisione del dominio e sullo scavo della geometria. Questo ha messo in luce che, nel caso di geometrie complesse, non è sempre possibile creare una mesh che si adatti perfettamente al dominio in esame ed è necessario trovare una via di mezzo tra le possibili strategie come evidenziato nel capitolo 5. Questo approccio ha portato ad un miglioramento dei risultati provenienti dalla simulazione e ha evidenziato come la mesh mista sia più valida, in questo caso, rispetto ad una completamente non strutturata.

Il modello realizzato potrà essere proficuamente adottato per future simulazioni termofluidodinamiche, con condizioni e passi temporali diversi; potrà essere migliorato sia a livello di raffinamento che come geometria permettendo ai futuri ricercatori di ottenere risultati ancora più precisi ed attendibili; ciò è stato possibile anche grazie ai programmi open-source, già citati nel capitolo 2, così da renderne possibile l'utilizzo a chiunque voglia. Ad esempio, oltre alla simulazione della turbolenza si potrà simulare l'effetto che il sodio che lambisce le pareti ha sull'aumento di temperatura di quest'ultime.

Riconoscimenti

L'attività sviluppata in questa tesi è stata supportata in parte dal programma di ricerca e formazione Euratom 2014-2018 col grant agreement n. 754501.

Acknowledgements

The activity developed in this thesis has been partially supported from the Euratom research and training programme 2014-2018 under the grant agreement n. 754501.

Bibliografia

- [1] *Enzo Zanchini*, Moto dei Fluidi e Termocinetica T.
- [2] *C. Geuzaine and J.-F. Remacle*. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 79(11), pp. 1309-1331, 2009. (<http://www.gmsh.info>).
- [3] *Python Software Foundation*. Python Language Reference version 3.6 disponibile su <http://www.python.org>.
- [4] *openFOAM*, disponibile su <http://www.openfoam.com>.
- [5] *Ahrens, James, Geveci, Berk, Law, Charles*, ParaView: An End-User Tool for Large Data Visualization, *Visualization Handbook*, Elsevier, 2005, ISBN-13: 978-0123875822 (<http://www.paraview.org>).
- [6] *John E. Kelly*, Generation IV International Forum: A decade of progress through international cooperation, 2014.
- [7] *J. Guidez, J. Bodi, K. Mikityuk, S. Beils*. Specification of the new system safety measures, 2018 (<http://www.esfr-smart.eu>).
- [8] *Andrei Rineiski, Clément Meriot, Christine Coquelet, Jiri Krepel, Emil Fridman, Konstantin Mikityuk* Specification of the new core safety measures, 2018 (<http://www.esfr-smart.eu>).
- [9] *Vitaly Sobolev*, Database of thermophysical properties of liquid metal coolants for GEN-IV, 2010.
- [10] *Sandia National Laboratories*, The Verdict Geometric Quality Library, 2007.