



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

**DIPARTIMENTO DI INGEGNERIA INDUSTRIALE (DIN)**

**CORSO DI LAUREA IN INGEGNERIA MECCANICA**

**SVILUPPO DI UN CODICE  
NUMERICO OPEN-SOURCE PER  
SIMULAZIONI FEM  
TERMOFLUIDODINAMICHE**

Tesi di laurea in Termofluidodinamica Applicata

**Relatore**

Prof. Massimiliano Rossi

**Presentata da**

Francesca Manzoni

**Correlatore**

Prof. Guglielmo Minelli

---

**Anno Accademico 2024/2025**



## Sommario

Il presente elaborato analizza, mediante simulazioni numeriche, una serie di fenomeni termo-fluidodinamici di complessità crescente, realizzati attraverso un ambiente di calcolo interamente basato su strumenti open-source.

Sono stati impiegati **Gmsh** per la generazione della geometria e della mesh, **FEniCSx** per la formulazione e la risoluzione delle equazioni agli elementi finiti (FEM) derivate dai bilanci di massa, quantità di moto ed energia, e **ParaView** per la post-elaborazione dei risultati.

L'attività di simulazione è stata sviluppata in modo progressivo. Inizialmente è stato analizzato il fenomeno della conduzione termica in una parete piana, sia in regime stazionario sia transitorio, al fine di valutare la diffusione del calore e la risposta termica del sistema. Successivamente è stato studiato il flusso laminare di Poiseuille in una sezione rettangolare, utilizzato come caso di riferimento per la validazione della formulazione variazionale e delle condizioni di moto.

Nella parte finale è stata considerata la cavità riscaldata, analizzata in regime stazionario e tempo-dipendente, con aria e acqua come fluidi di lavoro. Il confronto tra i due casi ha evidenziato l'influenza delle proprietà fisiche sui moti convettivi e sulla distribuzione della temperatura.

I risultati, interpretati mediante i numeri adimensionali di Reynolds e Rayleigh, hanno confermato la coerenza del modello numerico e la capacità dell'approccio adottato di descrivere correttamente i diversi regimi di moto. Le metodologie sviluppate costituiscono una base solida per estensioni future, quali lo studio e l'impiego di materiali a cambiamento di fase (PCM) per l'accumulo e la gestione dell'energia termica.

# Notazione e simboli

Tabella 1: Elenco dei simboli e delle grandezze fisiche utilizzate

| Simbolo      | Tipo          | Unità                                   | Significato                                         |
|--------------|---------------|-----------------------------------------|-----------------------------------------------------|
| $\mathbf{u}$ | Vettore       | $\text{m s}^{-1}$                       | Velocità del fluido                                 |
| $p$          | Scalare       | Pa                                      | Pressione meccanica (o modificata)                  |
| $T$          | Scalare       | K                                       | Temperatura                                         |
| $\rho_0$     | Scalare       | $\text{kg m}^{-3}$                      | Densità di riferimento (costante)                   |
| $\mu, \nu$   | Scalari       | $\text{Pa s}; \text{m}^2 \text{s}^{-1}$ | Viscosità dinamica e cinematica                     |
| $k$          | Scalare       | $\text{W m}^{-1} \text{K}^{-1}$         | Conducibilità termica                               |
| $c_p$        | Scalare       | $\text{J kg}^{-1} \text{K}^{-1}$        | Calore specifico a pressione costante               |
| $\beta$      | Scalare       | $\text{K}^{-1}$                         | Coefficiente di dilatazione termica volumetrico     |
| $\mathbf{g}$ | Vettore       | $\text{m s}^{-2}$                       | Accelerazione di gravità ( $\mathbf{g} = (0, -g)$ ) |
| $g$          | Scalare       | $\text{m s}^{-2}$                       | Modulo dell'accelerazione di gravità                |
| $\alpha$     | Scalare       | $\text{m}^2 \text{s}^{-1}$              | Diffusività termica $\alpha = \frac{k}{\rho_0 c_p}$ |
| $U, L$       | Scalari       | $\text{m s}^{-1}; \text{m}$             | Velocità e lunghezza caratteristiche                |
| Re, Pr, Ra   | Adimensionali | —                                       | Numeri di Reynolds, Prandtl, Rayleigh               |

| Simbolo                                                | Tipo             | Unità                              | Significato                                                                             |
|--------------------------------------------------------|------------------|------------------------------------|-----------------------------------------------------------------------------------------|
| $\Delta T$                                             | Scalare          | K                                  | Salto di temperatura caratteristico                                                     |
| $T_{\text{hot}}, T_{\text{cold}}, T_{\text{ref}}, T_0$ | Scalari          | K                                  | Temperature imposte, di riferimento e iniziale                                          |
| $\mathbf{f}, \mathbf{f}_b$                             | Vettori          | $\text{m s}^{-2}; \text{N m}^{-3}$ | Forze di corpo e forza di galleggiamento (Boussinesq)                                   |
| $\mathbf{D}(\mathbf{u})$                               | Operatore        | –                                  | Tensore delle deformazioni: $\frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top)$ |
| $\tau_{\text{SUPG}}$                                   | Parametro        | s                                  | Parametro di stabilizzazione SUPG per il termine convettivo                             |
| $q, \mathbf{v}, s$                                     | Funzioni di test | –                                  | Funzioni di prova per pressione, velocità e temperatura                                 |
| $V, Q, W$                                              | Spazi            | –                                  | Spazi funzionali delle incognite                                                        |
| $V_0, W_0$                                             | Spazi            | –                                  | Spazi di test con condizioni di Dirichlet                                               |
| $\bar{\mathbf{t}}, \bar{q}$                            | Dati imposti     | $\text{N m}^{-2}; \text{W m}^{-2}$ | Trazione e flusso termico prescritti                                                    |
| $\Delta t, t^n, t^{n+1}$                               | Tempo            | s                                  | Passo temporale e istanti discreti                                                      |
| $\mathbf{u}^*, \phi$                                   | Campi            | $\text{m s}^{-1}; \text{Pa}$       | Velocità provvisoria e correzione di pressione                                          |
| $p_{\text{in}}, p_{\text{out}}$                        | Scalari          | Pa                                 | Pressioni ai contorni (flusso di Poiseuille)                                            |
| $\Gamma_{\text{hot}}$                                  | Bordi            | –                                  | Parete calda ( $T = T_{\text{hot}}$ )                                                   |
| $\Gamma_{\text{cold}}$                                 | Bordi            | –                                  | Parete fredda ( $T = T_{\text{cold}}$ )                                                 |
| $\Gamma_{\text{adiab}}$                                | Bordi            | –                                  | Pareti adiabatiche ( $\nabla T \cdot \mathbf{n} = 0$ )                                  |
| $\Gamma_D$                                             | Bordi            | –                                  | Bordo con condizioni di Dirichlet generiche                                             |
| $\Gamma_N$                                             | Bordi            | –                                  | Bordo con condizioni di Neumann generiche                                               |

| Simbolo          | Tipo    | Unità | Significato                         |
|------------------|---------|-------|-------------------------------------|
| $\mathbf{n}$     | Vettore | –     | Versore normale uscente al contorno |
| $\Omega$         | Dominio | –     | Dominio computazionale              |
| $\partial\Omega$ | Bordo   | –     | Frontiera del dominio $\Omega$      |

# Indice

|          |                                                 |           |
|----------|-------------------------------------------------|-----------|
| <b>1</b> | <b>Introduzione</b>                             | <b>6</b>  |
| <b>2</b> | <b>Fondamenti teorici</b>                       | <b>8</b>  |
| 2.1      | Equazioni di riferimento . . . . .              | 8         |
| 2.1.1    | Equazioni fondamentali . . . . .                | 8         |
| 2.1.2    | Numeri adimensionali e regime di moto . . . . . | 11        |
| 2.1.3    | Regime laminare nelle simulazioni . . . . .     | 11        |
| 2.2      | Condizioni tipiche . . . . .                    | 12        |
| 2.3      | Formulazione debole . . . . .                   | 12        |
| 2.3.1    | Procedura generale . . . . .                    | 12        |
| 2.3.2    | Equazione di continuità . . . . .               | 16        |
| 2.3.3    | Equazioni di Navier–Stokes . . . . .            | 16        |
| 2.3.4    | Equazione dell’energia . . . . .                | 16        |
| 2.3.5    | Discretizzazione temporale . . . . .            | 17        |
| 2.3.6    | Sintesi del problema accoppiato . . . . .       | 18        |
| <b>3</b> | <b>Metodi</b>                                   | <b>19</b> |
| 3.1      | Metodi . . . . .                                | 19        |
| 3.1.1    | Gmsh . . . . .                                  | 19        |
| 3.1.2    | FEniCSx . . . . .                               | 20        |
| 3.1.3    | ParaView . . . . .                              | 21        |
| <b>4</b> | <b>Risultati</b>                                | <b>23</b> |
| 4.1      | Conduzione termica . . . . .                    | 23        |

|                     |                                                               |           |
|---------------------|---------------------------------------------------------------|-----------|
| 4.1.1               | Problema stazionario . . . . .                                | 24        |
| 4.1.2               | Problema transitorio . . . . .                                | 25        |
| 4.2                 | Flusso laminare di Poiseuille . . . . .                       | 26        |
| 4.3                 | Cavità riscaldata . . . . .                                   | 28        |
| 4.3.1               | Problema stazionario . . . . .                                | 29        |
| 4.3.2               | Problema tempo-dipendente . . . . .                           | 31        |
| <b>5</b>            | <b>Conclusioni</b>                                            | <b>37</b> |
| <b>A</b>            | <b>Appendice</b>                                              | <b>39</b> |
| A.1                 | Introduzione . . . . .                                        | 39        |
| A.1.1               | Prerequisiti . . . . .                                        | 39        |
| A.2                 | Caso di studio . . . . .                                      | 40        |
| A.2.1               | Condizioni al contorno . . . . .                              | 40        |
| A.3                 | Creazione della mesh su Gmsh . . . . .                        | 40        |
| A.4                 | Creazione del DockerFile . . . . .                            | 42        |
| A.5                 | Docker Desktop . . . . .                                      | 43        |
| A.6                 | File Python . . . . .                                         | 43        |
| A.6.1               | Importazione delle librerie . . . . .                         | 44        |
| A.6.2               | Importazione della mesh da Gmsh . . . . .                     | 45        |
| A.6.3               | Definizione delle costanti e dei parametri numerici . . . . . | 45        |
| A.6.4               | Costruzione degli spazi funzionali . . . . .                  | 46        |
| A.6.5               | Condizioni iniziali e al contorno . . . . .                   | 47        |
| A.6.6               | Formulazione variazionale . . . . .                           | 47        |
| A.6.7               | Integrazione temporale . . . . .                              | 48        |
| A.6.8               | Esportazione dei risultati . . . . .                          | 49        |
| A.7                 | Esecuzione della simulazione in ambiente Docker . . . . .     | 49        |
| A.8                 | Visualizzazione in Paraview . . . . .                         | 51        |
| A.9                 | Conclusione . . . . .                                         | 52        |
| <b>Bibliografia</b> |                                                               | <b>52</b> |



# Capitolo 1

## Introduzione

L'efficienza nei processi di accumulo e utilizzo dell'energia costituisce un tema di crescente interesse nell'ambito dell'ingegneria meccanica e termica.

In questo contesto si collocano i **materiali a cambiamento di fase (PCM)**, che consentono di **immagazzinare e rilasciare calore** durante la transizione di fase, ad esempio da solido a liquido. Tale caratteristica li rende idonei a favorire la **stabilizzazione del campo termico** e a migliorare la **distribuzione del calore** all'interno dei sistemi energetici.

L'impiego dei PCM presenta tuttavia alcune **limitazioni**: la **bassa conducibilità termica** rallenta i processi di fusione e solidificazione, mentre la **modellazione numerica** richiede una corretta gestione del **calore latente** e delle interazioni tra campo termico e moto del fluido.

Per ovviare a tali limiti, una delle possibili strategie di miglioramento è rappresentata dallo **streaming acustico**, un fenomeno di moto indotto da onde acustiche nel fluido. Attraverso l'applicazione di campi acustici opportunamente modulati, è possibile **incrementare il trasporto convettivo** all'interno del PCM, favorendo una fusione più omogenea e rapida. L'interazione tra onde acustiche e campo termico rappresenta dunque un approccio per **aumentare l'efficienza di scambio termico** e migliorare le prestazioni complessive dei sistemi di accumulo energetico.

La presente tesi si propone di **sviluppare e validare codici numerici** per la simulazione di fenomeni termo-fluidodinamici, utilizzando strumenti open-source come **Gmsh** per la generazione della geometria e della mesh, **FEniCSx/Dolfinx** per la risoluzione delle equazioni differenziali e **ParaView** per l'analisi e la visualizzazione dei risultati.

L'impiego di codici open-source permette di disporre di una piattaforma di calcolo **flessi-**

**bile ed economicamente accessibile**, caratteristica che ne rende l'utilizzo particolarmente vantaggioso in ambito accademico e di ricerca.

Inoltre, la possibilità di sviluppare e modificare direttamente le equazioni offre un **controllo completo sui modelli fisici e numerici**, consentendo di adattarli a fenomeni complessi come quelli che intervengono nei processi di convezione.

Il presente lavoro si concentra sulla **modellazione numerica del trasporto di quantità di moto, del campo termico e della condizione di incomprimibilità**, con l'obiettivo di analizzare i principali comportamenti fluidodinamici e termici in domini bidimensionali.

Gli **sviluppi futuri** prevedono l'**applicazione dei codici sviluppati ai materiali a cambiamento di fase**, al fine di estendere il modello attuale e analizzare in modo più approfondito i processi di fusione e solidificazione accoppiati alla convezione naturale.

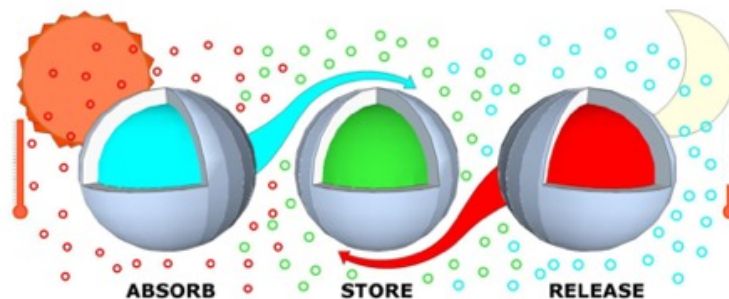


Figura 1.1: Schema rappresentativo del funzionamento di un materiale a cambiamento di fase (PCM). Durante la fase di *assorbimento* il materiale fonde accumulando calore latente; nella fase di *accumulo* l'energia rimane immagazzinata nella massa fusa; infine, nella fase di *rilascio*, il materiale solidifica restituendo l'energia termica all'ambiente circostante.

# Capitolo 2

## Fondamenti teorici

### 2.1 Equazioni di riferimento

Lo studio dei fenomeni termo-fluidodinamici richiede la definizione delle equazioni fondamentali che governano il moto di un fluido e l'evoluzione di temperatura, pressione e velocità al suo interno.

#### 2.1.1 Equazioni fondamentali

Le equazioni che descrivono il comportamento dei fluidi e il trasporto termico derivano dai principi di conservazione di massa, quantità di moto ed energia. Nel seguito vengono riportate nella loro forma generale e, dove opportuno, nella forma semplificata per fluidi incomprimibili, adottata in tutte le simulazioni di questa tesi.

#### Equazione di continuità

**Caso generale.** Il principio di conservazione della massa, per un volume di controllo  $V \subset \Omega$  con superficie  $S = \partial V$ , si scrive come:

$$\frac{d}{dt} \int_V \rho dV + \int_S \rho \mathbf{u} \cdot \mathbf{n} dS = 0.$$

Applicando il teorema della divergenza e portando il bilancio in forma locale, si ottiene:

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

**Caso incomprimibile.** Assumendo densità costante  $\rho = \rho_0 = \text{costante}$ , la relazione si riduce a:

$$\nabla \cdot \mathbf{u} = 0,$$

che rappresenta il vincolo di incomprimibilità.

—

## Equazioni di Navier–Stokes

**Caso generale.** Il bilancio della quantità di moto, derivato dalla seconda legge di Newton, è espresso da:

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f},$$

dove  $\mathbf{u}$  è il campo di velocità,  $p$  la pressione meccanica,  $\mu$  la viscosità dinamica e  $\mathbf{f}$  il vettore delle forze di corpo (ad esempio la gravità).

**Caso incomprimibile.** Per densità costante ( $\rho = \rho_0$ ) e fluido newtoniano, il sistema assume la forma:

$$\begin{cases} \rho_0 \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho_0 \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0. \end{cases}$$

## Approssimazione di Boussinesq

**Caso generale.** Nei fenomeni di convezione naturale, la densità varia con la temperatura secondo la legge lineare:

$$\rho(T) = \rho_0 [1 - \beta(T - T_{\text{ref}})],$$

dove  $\rho_0$  è la densità a temperatura di riferimento  $T_{\text{ref}}$  e  $\beta$  il coefficiente di dilatazione termica volumetrico.

**Caso incomprimibile con galleggiamento.** In tutti i termini di Navier–Stokes si assume  $\rho = \rho_0$  costante, eccetto nella forza peso, dove la variazione di densità produce una forza di galleggiamento:

$$\rho \mathbf{g} \approx \rho_0 \mathbf{g} - \rho_0 \beta (T - T_{\text{ref}}) \mathbf{g}.$$

Il termine costante  $\rho_0 \mathbf{g}$  viene assorbito nella pressione modificata, e rimane il contributo di galleggiamento:

$$\mathbf{f}_b = -\rho_0 \beta (T - T_{\text{ref}}) \mathbf{g}.$$

*Convenzione:*  $\mathbf{g} = (0, -g)$ , cioè diretta verso il basso. In tal modo, quando  $T > T_{\text{ref}}$  la spinta di galleggiamento risulta diretta verso l'alto.

L'approssimazione di Boussinesq è valida se  $\beta \Delta T \ll 1$ , condizione rispettata per fluidi comuni in intervalli moderati di temperatura.

## Equazione dell'energia

**Caso generale.** Il bilancio dell'energia interna per un fluido newtoniano con proprietà costanti, includendo i termini dissipativi e le sorgenti volumetriche, è dato da:

$$\rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = \nabla \cdot (k \nabla T) + \Phi + \dot{q}_v,$$

dove  $c_p$  è il calore specifico,  $k$  la conducibilità termica,  $\Phi$  la dissipazione viscosa e  $\dot{q}_v$  una sorgente volumetrica di calore.

**Caso incomprimibile semplificato.** Trascurando dissipazione viscosa e sorgenti di calore:

$$\rho_0 c_p \left( \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = k \nabla^2 T.$$

Il termine a sinistra rappresenta il trasporto di calore (conduzione e convezione), mentre il termine a destra descrive la diffusione termica secondo la legge di Fourier.

### 2.1.2 Numeri adimensionali e regime di moto

L'analisi dei numeri adimensionali consente di caratterizzare i fenomeni fisici e di distinguere i diversi regimi di flusso. Le grandezze principali sono:

- **Numero di Reynolds**

$$Re = \frac{UL}{\nu}$$

con  $U$  velocità caratteristica [ $\text{m s}^{-1}$ ],  $L$  lunghezza caratteristica [ $\text{m}$ ],  $\nu$  viscosità cinematica [ $\text{m}^2 \text{s}^{-1}$ ].

Indica il rapporto tra effetti inerziali e viscosi.

*Bassi valori*  $\Rightarrow$  moto laminare

*alti valori*  $\Rightarrow$  possibile transizione a turbolenza.

- **Numero di Prandtl**

$$Pr = \frac{\nu}{\alpha}$$

con  $\alpha = k/(\rho_0 c_p)$  diffusività termica [ $\text{m}^2 \text{s}^{-1}$ ].

Esprime il rapporto tra diffusività di quantità di moto e diffusività termica.

- **Numero di Rayleigh**

$$Ra = \frac{g \beta \Delta T L^3}{\nu \alpha}$$

con  $g$  gravità [ $\text{m s}^{-2}$ ],  $\Delta T$  differenza di temperatura [ $\text{K}$ ].

Governa la convezione naturale: valori bassi  $\Rightarrow$  conduzione prevalente

valori alti  $\Rightarrow$  convezione più intensa e possibile instabilità.

### 2.1.3 Regime laminare nelle simulazioni

Nel contesto di questa tesi si è scelto di lavorare in regime laminare. Le motivazioni principali sono:

- i numeri di Reynolds e di Rayleigh considerati restano in intervalli per cui non si sviluppa turbolenza macroscopica;

- il regime laminare è coerente con le ipotesi di base dei modelli numerici implementati, semplificando la formulazione e riducendo la necessità di modelli di turbolenza;
- la mesh e i tempi di calcolo rimangono gestibili senza introdurre ulteriori incertezze modellistiche.

## 2.2 Condizioni tipiche

Nelle simulazioni presentate si assumono:

- proprietà termofisiche costanti ( $\rho_0, c_p, k$ );
- assenza di dissipazione e sorgenti ( $\Phi = 0, \dot{q}_v = 0$ );
- pareti isoterme ( $T = \bar{T}$ ) o adiabatiche ( $\nabla T \cdot \mathbf{n} = 0$ );
- assenza di sorgenti di calore
- fluido inizialmente a temperatura uniforme  $T_0$  e velocità nulla.

Si assume  $T_{\text{ref}} = T_0$  salvo diversa indicazione.

## 2.3 Formulazione debole

Le equazioni in forma differenziale (*forma forte*) descrivono i fenomeni punto per punto, ma richiedono soluzioni molto regolari (ad esempio derivate seconde continue). Il metodo degli elementi finiti utilizza invece la **formulazione debole**, che riduce i requisiti di regolarità.

### 2.3.1 Procedura generale

Il punto di partenza di ogni problema fisico espresso mediante equazioni alle derivate parziali è la sua **forma forte**, ossia l'espressione differenziale che deve essere soddisfatta in ogni punto del dominio.

Tuttavia, le soluzioni di tali equazioni spesso non sono sufficientemente regolari da ammettere derivate di ordine elevato, rendendo complessa una risoluzione numerica diretta.

Per superare questa limitazione, il metodo degli elementi finiti utilizza la **forma debole**, che consiste nel riscrivere il problema in forma integrale, riducendo l'ordine delle derivate richieste. Il passaggio dalla forma forte alla forma debole segue i seguenti passaggi fondamentali:

1. moltiplicare ciascuna equazione per una funzione di test appartenente allo spazio opportuno;
2. integrare su tutto il dominio  $\Omega$ ;
3. applicare il teorema della divergenza per ridurre l'ordine delle derivate;
4. imporre le condizioni al contorno di Dirichlet e di Neumann.

Una volta riscritta l'equazione in forma integrale, è necessario specificare gli spazi funzionali in cui cercare la soluzione e quelli in cui le funzioni di test sono definite. A tale scopo si introducono due categorie di funzioni:

### **Trial functions (funzioni incognite)**

Rappresentano le grandezze fisiche che si vogliono determinare, cioè la soluzione del problema.

- **Velocità:**  $\mathbf{u} \in V$ ;
- **Pressione:**  $p \in Q$ ;
- **Temperatura:**  $T \in W$ .

Gli spazi  $V$ ,  $Q$  e  $W$  sono detti *spazi funzionali delle incognite* e contengono le funzioni che soddisfano le condizioni di regolarità del problema.

### **Test functions (funzioni di prova)**

Sono funzioni arbitrarie appartenenti agli stessi spazi delle incognite, ma nulle sul bordo dove sono imposte condizioni di Dirichlet. Servono per verificare la validità dell'equazione in senso integrale.

- **Velocità:**  $\mathbf{v} \in V_0$ ;



- **Pressione:**  $q \in Q$ ;
- **Temperatura:**  $s \in W_0$ .

Gli spazi  $V_0$  e  $W_0$  rappresentano sottoinsiemi di  $V$  e  $W$  che rispettano le condizioni al contorno di tipo Dirichlet (funzioni nulle sul bordo prescritto).

### Forme bilineari e lineari

La forma debole ottenuta con i passaggi precedenti contiene termini che coinvolgono le incognite e le funzioni di test. Per rappresentare questi contributi in modo sistematico, si introducono due operatori fondamentali:

- La **forma bilineare**, indicata con  $a(\cdot, \cdot)$ , raccoglie i termini che dipendono sia dalle incognite sia dalle funzioni di test. È detta “bilineare” perché è lineare rispetto a ciascuno dei due argomenti presi separatamente. Ad esempio:

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega.$$

Questo termine rappresenta, ad esempio, la diffusione in un problema termico o la viscosità in un problema fluido-dinamico.

- La **forma lineare**, indicata con  $L(\cdot)$ , contiene invece i termini noti del problema (sorgenti, forze di corpo o flussi imposti). Essa è lineare rispetto alla sola funzione di test:

$$L(v) = \int_{\Omega} f v \, d\Omega + \int_{\Gamma_N} g v \, d\Gamma.$$

Una volta definite queste forme, il problema variazionale può essere scritto come:

Trovare  $u \in V$  tale che  $a(u, v) = L(v)$  per ogni  $v \in V_0$ .

Questo passaggio è essenziale poiché consente di:

1. separare la parte del problema che dipende dalle incognite (forma bilineare) da quella che dipende dai dati noti (forma lineare);
2. tradurre la formulazione continua in un linguaggio algebrico, che può essere discretizzato numericamente.

## Discretizzazione temporale e forme dipendenti dal tempo

Quando il problema è transitorio, le variabili dipendono anche dal tempo  $t$ . Il dominio temporale viene quindi suddiviso in intervalli di ampiezza  $\Delta t$ , e le variabili sono valutate agli istanti discreti  $t^n = n \Delta t$ .

La derivata temporale è approssimata da uno schema numerico: lo **schema implicito di Eulero**:

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi^{n+1} - \phi^n}{\Delta t},$$

dove  $\phi$  rappresenta una grandezza generica, come la velocità  $\mathbf{u}$  o la temperatura  $T$ .

Con questa approssimazione, le forme bilineari e lineari diventano **dipendenti dal tempo**:

$$a^{n+1}(\cdot, \cdot) \quad L^{n+1}(\cdot),$$

e devono essere risolte ad ogni passo temporale.

Il problema discreto assume quindi la forma:

$$A^{n+1} \mathbf{x}^{n+1} = \mathbf{b}^{n+1},$$

dove:

- la matrice  $A^{n+1}$  deriva dalla forma bilineare  $a^{n+1}(\cdot, \cdot)$ ;
- il vettore  $\mathbf{b}^{n+1}$  deriva dalla forma lineare  $L^{n+1}(\cdot)$ ;
- la soluzione  $\mathbf{x}^{n+1}$  rappresenta le variabili incognite al tempo  $t^{n+1}$ .

In sintesi:

- la **forma debole** riduce l'ordine delle derivate e permette di formulare il problema in modo compatibile con gli spazi funzionali di Sobolev;
- le **forme bilineari e lineari** organizzano i contributi del problema in una struttura algebrica gestibile numericamente;
- la **discretizzazione temporale** consente di risolvere i problemi evolutivi passo per passo nel tempo.

### 2.3.2 Equazione di continuità

Forma forte

$$\nabla \cdot \mathbf{u} = 0.$$

Forma debole

$$\int_{\Omega} q (\nabla \cdot \mathbf{u}) d\Omega = 0, \quad \forall q \in Q.$$

Forme associate

$$c(\mathbf{u}, q) = \int_{\Omega} q (\nabla \cdot \mathbf{u}) d\Omega,$$
$$L_p(q) = 0.$$

### 2.3.3 Equazioni di Navier–Stokes

Forma forte

$$\rho_0 \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} - \rho_0 \beta (T - T_{\text{ref}}) \mathbf{g}.$$

*Nota:*  $p$  denota la pressione modificata, ottenuta assorbendo il termine idrostatico  $\rho_0 \mathbf{g}$ .

Forma debole

$$\int_{\Omega} \rho_0 \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) \cdot \mathbf{v} + \mu \nabla \mathbf{u} : \nabla \mathbf{v} - p (\nabla \cdot \mathbf{v}) d\Omega = - \int_{\Omega} \rho_0 \beta (T - T_{\text{ref}}) \mathbf{g} \cdot \mathbf{v} d\Omega + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma.$$

Forme associate

$$a_u(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \rho_0 (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} d\Omega + \int_{\Omega} \mu (\nabla \mathbf{u} : \nabla \mathbf{v}) d\Omega - \int_{\Omega} p (\nabla \cdot \mathbf{v}) d\Omega,$$

$$L_u(\mathbf{v}) = - \int_{\Omega} \rho_0 \beta (T - T_{\text{ref}}) \mathbf{g} \cdot \mathbf{v} d\Omega + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma.$$

### 2.3.4 Equazione dell'energia

Forma forte

$$\rho_0 c_p \left( \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = k \nabla^2 T.$$

### Forma debole

$$\int_{\Omega} \rho_0 c_p \left( \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) s \, d\Omega + \int_{\Omega} k (\nabla T \cdot \nabla s) \, d\Omega = \int_{\Gamma_q} \bar{q} s \, d\Gamma.$$

### Forme associate

$$a_T(T, s) = \int_{\Omega} \rho_0 c_p (\mathbf{u} \cdot \nabla T) s \, d\Omega + \int_{\Omega} k (\nabla T \cdot \nabla s) \, d\Omega,$$

$$L_T(s) = \int_{\Gamma_q} \bar{q} s \, d\Gamma.$$

### 2.3.5 Discretizzazione temporale

Per la risoluzione di problemi transitori, si introduce un passo temporale  $\Delta t$  e una sequenza di istanti discreti  $t^n = n \Delta t$ . Le derivate temporali vengono approssimate mediante lo **schema implicito di Eulero**:

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi^{n+1} - \phi^n}{\Delta t},$$

dove  $\phi$  rappresenta genericamente una variabile incognita, come la velocità  $\mathbf{u}$  o la temperatura  $T$ .

### Equazioni della quantità di moto (forma debole discreta)

$$\int_{\Omega} \rho_0 \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \cdot \mathbf{v} \, d\Omega + \int_{\Omega} \rho_0 (\mathbf{u}^n \cdot \nabla \mathbf{u}^{n+1}) \cdot \mathbf{v} \, d\Omega + \int_{\Omega} \mu (\nabla \mathbf{u}^{n+1} : \nabla \mathbf{v}) \, d\Omega - \int_{\Omega} p^{n+1} (\nabla \cdot \mathbf{v}) \, d\Omega = - \int_{\Omega} \rho_0 \beta (T^n - T_{\text{ref}}) \mathbf{g} \cdot \mathbf{v} \, d\Omega.$$

### Equazione dell'energia (forma debole discreta)

$$\int_{\Omega} \rho_0 c_p \frac{T^{n+1} - T^n}{\Delta t} s \, d\Omega + \int_{\Omega} \rho_0 c_p (\mathbf{u}^n \cdot \nabla T^{n+1}) s \, d\Omega + \int_{\Omega} k (\nabla T^{n+1} \cdot \nabla s) \, d\Omega = \int_{\Gamma_q} \bar{q} s \, d\Gamma.$$

### Quantità di moto in forma lineare e bilineare

$$a_u^{n+1}(\mathbf{u}^{n+1}, \mathbf{v}) = \frac{\rho_0}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} \, d\Omega + \int_{\Omega} \rho_0 (\mathbf{u}^n \cdot \nabla \mathbf{u}^{n+1}) \cdot \mathbf{v} \, d\Omega + \int_{\Omega} \mu (\nabla \mathbf{u}^{n+1} : \nabla \mathbf{v}) \, d\Omega - \int_{\Omega} p^{n+1} (\nabla \cdot \mathbf{v}) \, d\Omega,$$

$$L_u^{n+1}(\mathbf{v}) = \frac{\rho_0}{\Delta t} \int_{\Omega} \mathbf{u}^n \cdot \mathbf{v} \, d\Omega - \int_{\Omega} \rho_0 \beta (T^n - T_{\text{ref}}) \mathbf{g} \cdot \mathbf{v} \, d\Omega.$$

### Energia in forma lineare e bilinere

$$a_T^{n+1}(T^{n+1}, s) = \frac{\rho_0 c_p}{\Delta t} \int_{\Omega} T^{n+1} s \, d\Omega + \int_{\Omega} \rho_0 c_p (\mathbf{u}^n \cdot \nabla T^{n+1}) s \, d\Omega + \int_{\Omega} k (\nabla T^{n+1} \cdot \nabla s) \, d\Omega,$$

$$L_T^{n+1}(s) = \frac{\rho_0 c_p}{\Delta t} \int_{\Omega} T^n s \, d\Omega + \int_{\Gamma_q} \bar{q} s \, d\Gamma.$$

A ogni passo temporale, il problema variazionale si riduce alla risoluzione dei seguenti sistemi lineari:

$$A_u^{n+1} \mathbf{u}^{n+1} = \mathbf{b}_u^{n+1}, \quad A_T^{n+1} \mathbf{T}^{n+1} = \mathbf{b}_T^{n+1},$$

dove le matrici  $A_u^{n+1}$  e  $A_T^{n+1}$  derivano rispettivamente dalle forme bilineari  $a_u^{n+1}$  e  $a_T^{n+1}$ , mentre i vettori dei termini noti  $\mathbf{b}_u^{n+1}$  e  $\mathbf{b}_T^{n+1}$  derivano dalle forme lineari  $L_u^{n+1}$  e  $L_T^{n+1}$ .

### 2.3.6 Sintesi del problema accoppiato

In forma compatta, il sistema variazionale si scrive come:

$$a^{n+1}((\mathbf{u}, p, T), (\mathbf{v}, q, s)) = L^{n+1}(\mathbf{v}, q, s),$$

con:

$$a^{n+1} = a_u^{n+1} + c + a_T^{n+1}, \quad L^{n+1} = L_u^{n+1} + L_p + L_T^{n+1}.$$

Dopo la discretizzazione spaziale, si ottiene il sistema matriciale:

$$A^{n+1} \mathbf{x}^{n+1} = \mathbf{b}^{n+1},$$

risolto ad ogni passo temporale mediante i solutori di PETSc.

# Capitolo 3

## Metodi

### 3.1 Metodi

Le simulazioni termo-fluido-dinamiche sono state realizzate mediante un workflow completamente numerico basato su strumenti open-source.

L'intero processo comprende la generazione della geometria e della mesh, la risoluzione numerica delle equazioni alle derivate parziali e la successiva analisi dei risultati.

Il flusso di lavoro utilizzato è illustrato in Figura 3.1.



Figura 3.1: Workflow di simulazione numerica: Gmsh per la generazione della mesh, FEniCSx per la risoluzione delle equazioni e ParaView per il post-processing.

#### 3.1.1 Gmsh

La definizione della geometria e la generazione della mesh agli elementi finiti sono state eseguite con il software Gmsh.

Il programma consente la costruzione di domini bidimensionali o tridimensionali tramite interfaccia grafica o linguaggio di scripting `.geo`, permettendo un controllo accurato sulla discretizzazione.

A ciascun bordo e superficie del dominio sono stati assegnati specifici *tag* per identificare le diverse condizioni al contorno. Il file risultante, in formato `.msh`, è stato utilizzato come input per la fase di calcolo.

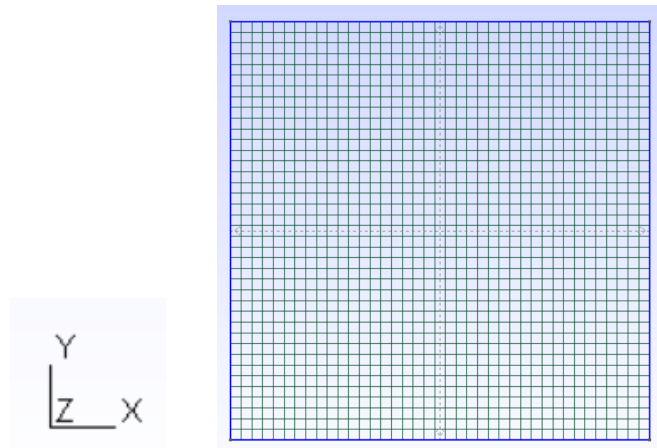


Figura 3.2: Esempio di mesh generata con **Gmsh** per la cavità quadrata. La discretizzazione è strutturata e composta da elementi quadrilateri di dimensione uniforme.

### 3.1.2 FEniCSx

La risoluzione numerica delle equazioni è stata condotta mediante il framework **FEniCSx**, basato sul metodo degli elementi finiti.

Il software consente la definizione diretta della formulazione variazionale dei problemi fisici tramite il linguaggio simbolico UFL (Unified Form Language).

Nel presente lavoro, **FEniCSx** è stato impiegato per:

- importare la mesh generata con **Gmsh** e costruire gli spazi funzionali per le variabili di campo;
- definire le proprietà fisiche del fluido (densità, viscosità, capacità termica, conducibilità);

- impostare le condizioni iniziali e al contorno;
- risolvere le equazioni di Navier–Stokes accoppiate con l’equazione dell’energia in regime stazionario o transitorio;
- esportare i risultati in formato `.xdmf`.

Le librerie PETSc e MPI sono state utilizzate per la gestione dei sistemi lineari e per l’esecuzione parallela del calcolo.

La Figura 3.3 riassume in forma concettuale la struttura del sistema di equazioni risolto con FEniCSx, evidenziando l’accoppiamento tra l’equazione dell’energia e le equazioni di Navier–Stokes, da cui derivano rispettivamente i campi di temperatura, velocità e pressione.

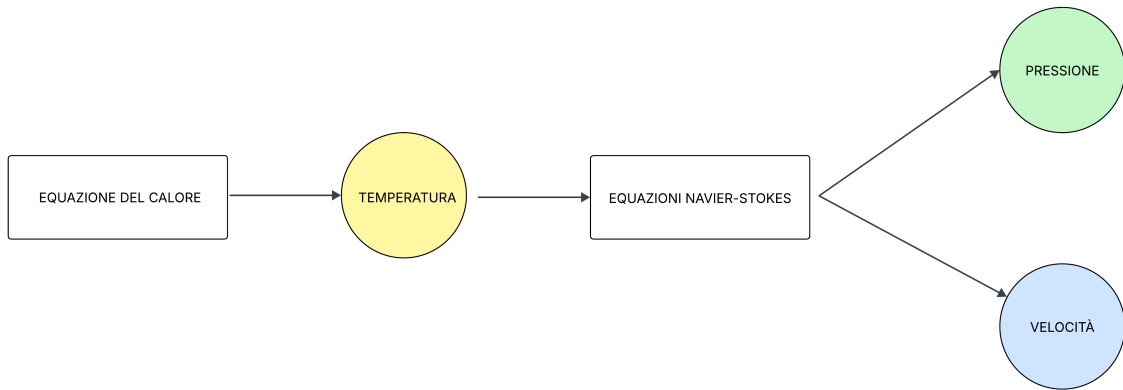


Figura 3.3: Schema concettuale del modello termo–fluidodinamico: la temperatura, ottenuta dalla risoluzione dell’equazione del calore, funge da sorgente termica per le equazioni di Navier–Stokes, dalle quali si ricavano i campi di pressione e velocità. Il workflow risulta applicabile esclusivamente a configurazioni non isoterme, in cui le variazioni di temperatura influenzano il moto del fluido.

### 3.1.3 ParaView

L’analisi e la visualizzazione dei risultati numerici sono state svolte con il software ParaView. Il programma consente l’elaborazione di dati scientifici in formato `.xdmf` e la rappresentazione di campi scalari e vettoriali. Le principali operazioni effettuate includono:

- la visualizzazione delle distribuzioni di temperatura e velocità del fluido;
- la generazione di linee di corrente e mappe di gradiente termico;



- la creazione di animazioni temporali per analisi transitorie;
- l'esportazione di immagini e video con sfondo uniforme per la documentazione dei risultati.

L'utilizzo combinato di **Gmsh**, **FEniCSx** e **ParaView** costituisce un ambiente integrato per la modellazione, la risoluzione e l'analisi di problemi di convezione naturale e di fenomeni termo-fluidodinamici complessi.

Un approfondimento dei codici sviluppati, delle condizioni al contorno e dei parametri numerici adottati è riportato in Appendice, dove vengono descritte in dettaglio le implementazioni utilizzate nelle diverse simulazioni.

# Capitolo 4

## Risultati

In questo capitolo vengono presentati i risultati numerici ottenuti dalle simulazioni condotte con il software **FEniCS**, utilizzando mesh generate in **Gmsh** e visualizzazioni in **ParaView**. Gli studi si concentrano su fenomeni di conduzione termica, flussi laminari e convezione naturale, con complessità crescente.

Per ciascun caso vengono descritte la geometria del dominio, le condizioni al contorno, le ipotesi fisiche adottate e i risultati numerici ottenuti.

### 4.1 Conduzione termica

La **conduzione** rappresenta uno dei principali meccanismi di trasferimento del calore e avviene all'interno di un corpo per effetto dei **gradienti di temperatura**, in assenza di moto macroscopico di materia.

Il fenomeno è legato ai processi di **diffusione energetica** e di **collisioni molecolari**, e risulta governato dalle proprietà termiche del materiale. La conduzione costituisce un caso di studio fondamentale per la comprensione dei processi di trasporto energetico e viene frequentemente utilizzata per la **validazione di modelli numerici** e l'analisi del comportamento termico dei sistemi.

Le simulazioni sviluppate hanno riguardato il comportamento conduttivo di un dominio **trapezoidale bidimensionale**, caratterizzato dalle seguenti dimensioni geometriche:

- base maggiore: 8 m;

- base minore: 4 m;
- altezza: 6 m.

Il dominio è stato discretizzato mediante una mesh strutturata composta da  $40 \times 40$  elementi quadrati. In tutte le analisi è stata assunta una **conducibilità termica unitaria**. Nei paragrafi seguenti sono presentati i risultati ottenuti nei casi stazionario e transitorio.

#### 4.1.1 Problema stazionario

La prima analisi riguarda un problema di **conduzione termica stazionaria** in un dominio trapezoidale bidimensionale. Si considerano le seguenti condizioni al contorno:

- **Parete sinistra:** temperatura costante  $T = 300$  K;
- **Parete destra:** temperatura costante  $T = 600$  K;
- **Pareti superiore e inferiore:** condizione di adiabaticità ( $\mathbf{n} \cdot \nabla T = 0$ ).

Il problema matematico è descritto dall'equazione di Laplace:

$$-\nabla \cdot (\nabla T) = 0 \quad \text{in } \Omega,$$

che rappresenta il bilancio del calore in assenza di sorgenti interne e di dipendenza temporale.

Il risultato numerico (Figura 4.1) mostra un profilo di temperatura regolare tra le due pareti isoterme, con andamento quasi lineare e gradiente costante.

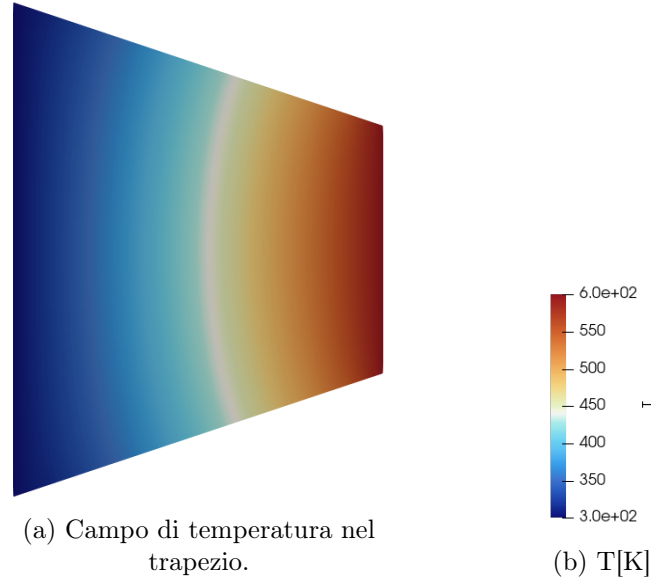


Figura 4.1: Simulazione stazionaria nel dominio trapezoidale.

#### 4.1.2 Problema transitorio

La seconda analisi prende in esame la **conduzione termica non stazionaria**, in cui la temperatura varia nel tempo fino al raggiungimento dell'equilibrio. Il fenomeno è descritto dall'equazione del calore:

$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0 \quad \text{in } \Omega \times (0, t_f],$$

Al tempo iniziale ( $t = 0$ ) l'intero dominio si trova a temperatura uniforme  $T(x, y, 0) = 300$  K. Per tempi successivi ( $t > 0$ ) vengono applicate le seguenti condizioni al contorno:

- **Parete sinistra:**  $T = 300$  K;
- **Parete destra:**  $T = 600$  K;
- **Pareti superiore e inferiore:** adiabatiche ( $\mathbf{n} \cdot \nabla T = 0$ ).

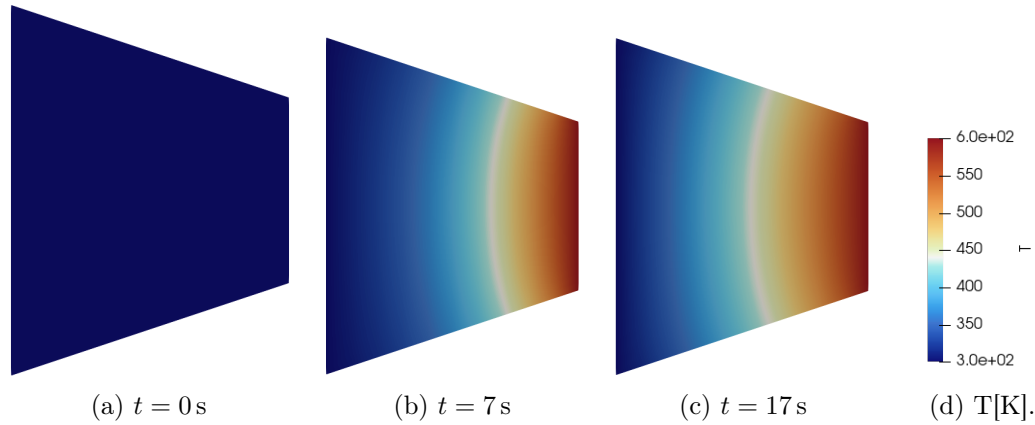


Figura 4.2: Evoluzione del campo di temperatura nel tempo.

La Figura 4.2 illustra l'evoluzione temporale del campo di temperatura: il calore si propaga gradualmente dalla parete destra, più calda, verso quella sinistra, fino al raggiungimento del regime stazionario.

Il profilo termico risulta coerente con la soluzione teorica prevista per il fenomeno diffusivo.

## 4.2 Flusso laminare di Poiseuille

Il **flusso di Poiseuille** descrive il moto laminare di un fluido viscoso confinato all'interno di un condotto, generato da una differenza di pressione applicata lungo la direzione principale del moto.

In tali condizioni, il profilo di velocità risulta regolare e simmetrico, con valori nulli sulle pareti a causa della condizione di aderenza e massimi nella regione centrale del dominio.

Questo tipo di flusso costituisce un caso di riferimento fondamentale per la **validazione di modelli numerici** e per l'analisi degli effetti viscosi nei moti confinati.

Nel presente studio è stata realizzata una **simulazione numerica del flusso di Poiseuille** al fine di verificare la corretta riproduzione del campo di moto e confrontare i risultati ottenuti con il comportamento teorico previsto.

La configurazione geometrica adottata consiste in una **sezione rettangolare di condotto** di lunghezza 4 m e altezza 1 m, rappresentativa di una sezione di **canale piano**.

Il dominio è stato discretizzato mediante una mesh strutturata composta da  $40 \times 10$  elementi quadrati. Il flusso è considerato **laminare e stazionario**.

Le condizioni al contorno imposte sono:

- **Inlet:** pressione assegnata  $p_{\text{in}} = 8.0 \text{ bar}$ ;
- **Outlet:** pressione assegnata  $p_{\text{out}} = 0.0 \text{ bar}$ ;
- **Pareti:** condizione di no-slip  $\mathbf{u} = \mathbf{0} \text{ [m/s]}$ .

Il problema è descritto dalle equazioni di Stokes per un fluido incomprimibile:

$$\begin{cases} -\nabla p + \mu \nabla^2 \mathbf{u} = \mathbf{0}, \\ \nabla \cdot \mathbf{u} = 0. \end{cases}$$

La soluzione numerica evidenzia un profilo di velocità parabolico nella sezione e una distribuzione lineare della pressione lungo la direzione del moto, in accordo con la teoria analitica. I risultati sono mostrati in Figura 4.3 e Figura 4.4.

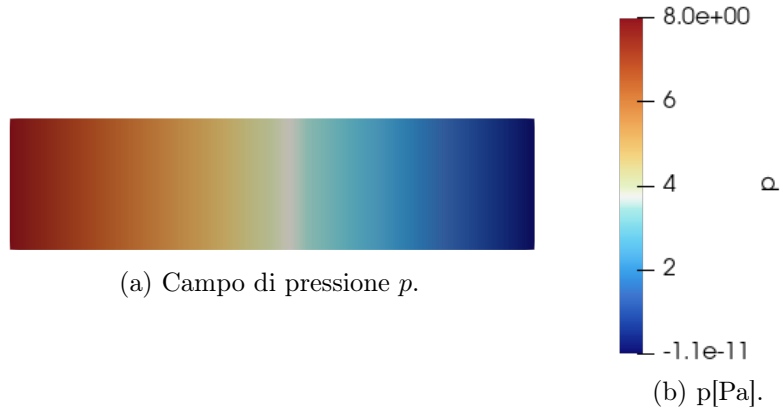


Figura 4.3: Distribuzione di pressione nel canale.

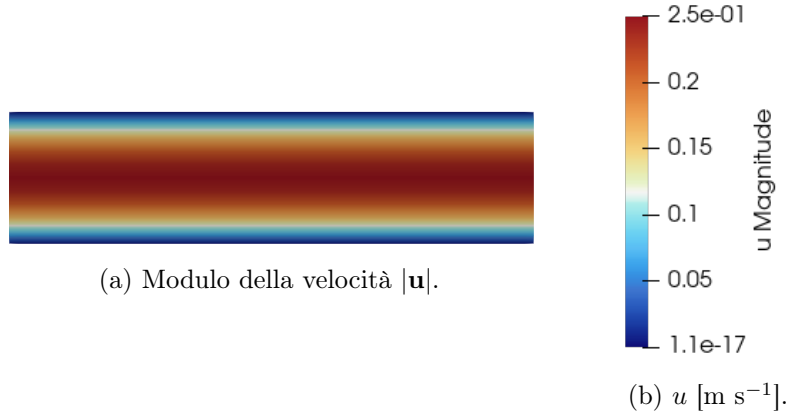


Figura 4.4: Distribuzione di velocità nel canale.

### 4.3 Cavit  riscaldata

Il problema della **cavit  riscaldata** rappresenta un caso di riferimento nello studio dei fenomeni di **convezione naturale** in domini chiusi.

La differenza di temperatura imposta tra due pareti opposte genera variazioni di densit  che inducono il moto del fluido, accoppiando cos  il campo di velocit  con quello termico. La modellazione del fenomeno   condotta nell'ambito dell'**approssimazione di Boussinesq**, che consente di assumere il fluido incomprimibile e di considerare le variazioni di densit  unicamente nel termine di galleggiamento legato alla forza di gravit .

Questa configurazione permette di analizzare in modo completo l'interazione tra il trasporto di calore e il moto convettivo, evidenziando i principali meccanismi di convezione naturale.

In questa simulazione sono stati utilizzati due diversi fluidi **aria** e **acqua** al fine di analizzare come le loro propriet  fisiche influenzino i moti convettivi e la distribuzione di temperatura all'interno del dominio.

Per comprendere meglio le differenze nel comportamento termo-fluidodinamico osservate tra i due casi, nella Tabella 4.1 sono riportate le principali propriet  fisiche dei fluidi impiegati nelle simulazioni.

Poich  il fluido   considerato incomprimibile, la pressione non costituisce una variabile di interesse ai fini dell'analisi; pertanto, nei risultati seguenti vengono riportati unicamente i campi di temperatura e velocit .

| Parametri dell'aria                 |                      |
|-------------------------------------|----------------------|
| Grandezza                           | Valore               |
| Densità $\rho$ [kg/m <sup>3</sup> ] | 1.225                |
| Viscosità dinamica $\mu$ [Pa·s]     | $1.8 \times 10^{-5}$ |
| Conducibilità termica $k$ [W/m·K]   | 0.0262               |
| Calore specifico $c_p$ [J/kg·K]     | 1005                 |
| Espansione termica $\beta$ [1/K]    | $3.4 \times 10^{-3}$ |

| Parametri dell'acqua                |                      |
|-------------------------------------|----------------------|
| Grandezza                           | Valore               |
| Densità $\rho$ [kg/m <sup>3</sup> ] | 997                  |
| Viscosità dinamica $\mu$ [Pa·s]     | $1.0 \times 10^{-3}$ |
| Conducibilità termica $k$ [W/m·K]   | 0.6                  |
| Calore specifico $c_p$ [J/kg·K]     | 4182                 |
| Espansione termica $\beta$ [1/K]    | $2.1 \times 10^{-4}$ |

Tabella 4.1: Proprietà fisiche dei fluidi utilizzati nelle simulazioni termo-fluidodinamiche.

#### 4.3.1 Problema stazionario

Nel primo caso si considera un regime **stazionario**, in cui il campo di moto e quello termico non variano nel tempo.

Il dominio di interesse è un quadrato di lato 10cm riempito di aria i cui parametri isici utilizzati sono riportati in Tabella 4.1.

La geometria è stata discretizzata mediante una mesh strutturata composta da  $40 \times 40$  elementi quadrati. Le condizioni al contorno applicate sono le seguenti:

- **Parete sinistra:**  $T_{\text{cold}} = 293 \text{ K}$ ;
- **Parete destra:**  $T_{\text{hot}} = 303 \text{ K}$ ;
- **Pareti superiore e inferiore:** adiabatiche ( $\mathbf{n} \cdot \nabla T = 0$ );
- **Tutte le pareti:** condizione di no-slip  $\mathbf{u} = \mathbf{0}$ .



Il sistema di equazioni risolto accoppia le equazioni di Navier–Stokes con quella dell’energia termica secondo l’approssimazione di Boussinesq:

$$\begin{cases} \rho_0 \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho_0 \beta (T - T_{\text{ref}}) \mathbf{g}, \\ \nabla \cdot \mathbf{u} = 0, \\ \nabla \cdot (k \nabla T) = 0. \end{cases}$$

Il risultato numerico mostra la formazione di una **cella convettiva** stabile, caratterizzata da moto ascendente lungo la parete calda e discendente lungo quella fredda, come illustrato in Figura 4.5.

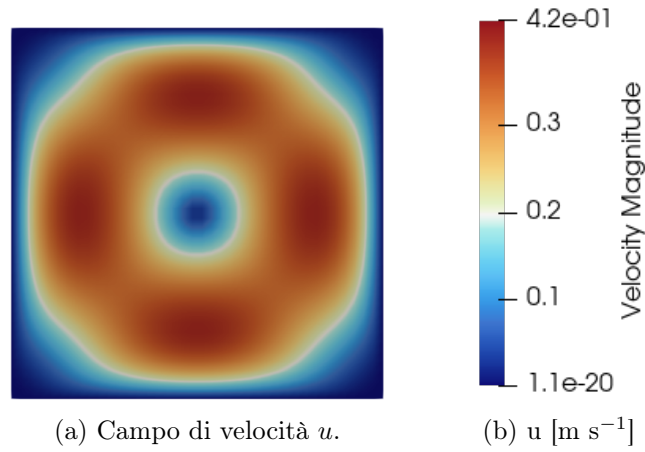


Figura 4.5: Distribuzione di velocità nella cavità.

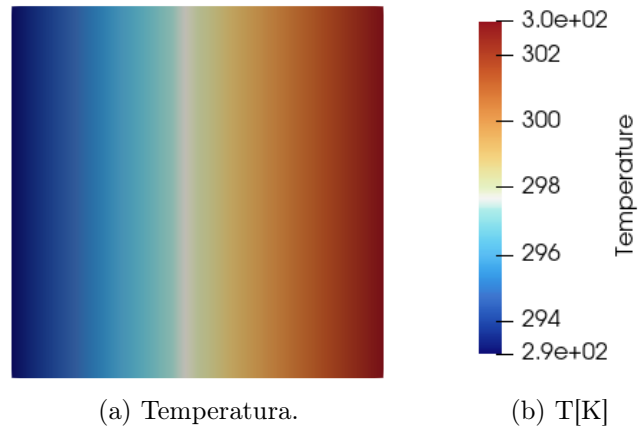


Figura 4.6: Distribuzione della temperatura

### 4.3.2 Problema tempo-dipendente

L'ultima analisi estende lo studio precedente al caso **non stazionario**, introducendo la dipendenza temporale dei campi fluidodinamici e termici.

La stessa geometria è stata discretizzata mediante una mesh più fine, composta da  $100 \times 100$  elementi, al fine di ottenere una rappresentazione più accurata dei gradienti di temperatura e velocità e ridurre l'errore numerico.

#### Condizioni iniziali

All'istante iniziale ( $t = 0$ ) la cavità si trova nella condizione di quiete:

- Temperatura iniziale:  $T_0 = 293 \text{ K}$ ;
- Velocità iniziale:  $\mathbf{u}_0 = \mathbf{0} \text{ m/s}$ ;
- Pressione iniziale:  $p_0 = 0 \text{ Pa}$ .

#### Condizioni per tempi successivi

Per tempi successivi ( $t > 0$ ) vengono imposte le seguenti condizioni:

- Parete inferiore riscaldata a temperatura costante:  $T_{\text{hot}} = 313 \text{ K}$ .

## Condizioni al contorno

Le condizioni al contorno sono definite come segue:

- Pareti superiore, sinistra e destra: adiabatiche  $\left(\frac{\partial T}{\partial n} = 0\right)$ ;
- Tutte le pareti: condizione di aderenza  $\mathbf{u} = \mathbf{0}$ .

Il problema è risolto mediante l'approssimazione di **Boussinesq**, che consente di modellare gli effetti delle variazioni di temperatura sulla densità e, di conseguenza, sul campo di pressione e sul moto convettivo.

I risultati mostrano, per diversi istanti temporali, l'evoluzione del campo termico e fluidodinamico: la formazione dello **strato termico** alla base, la **plume calda** ascendente e l'instaurarsi della circolazione convettiva all'interno della cavità.

## Confronto tra aria e acqua

Sono state considerate due configurazioni per analizzare l'influenza del fluido e della scala geometrica:

- Cavità quadrata di lato  $L = 10$  cm, riempita di **aria**
- Cavità quadrata di lato  $L = 2$  cm, riempita di **acqua**.

Le caratteristiche fisiche dei due fluidi sono descritte nella Tabella [4.1](#).

Si osserva come i valori dei numeri adimensionali risultino dello stesso ordine di grandezza per entrambi i casi:

$$\begin{array}{ll} \text{Aria:} & Re \approx 8.6 \times 10^2, \quad Ra \approx 5.26 \times 10^5 \\ \text{Acqua:} & Re \approx 2.83 \times 10^2, \quad Ra \approx 5.62 \times 10^5 \end{array}$$

Il numero di **Reynolds** è stato calcolato direttamente all'interno delle simulazioni, ricavando la velocità caratteristica dai valori medi del campo di moto e utilizzando i parametri fisici del fluido (densità, viscosità dinamica e dimensione caratteristica) impostati nel modello numerico. I valori ottenuti sono stati verificati attraverso i **print** del codice, in modo da garantire la coerenza con le condizioni al contorno e le proprietà del fluido considerate.

Entrambi i fluidi mostrano valori di **Rayleigh** molto simili, indice del fatto che, nonostante la diversa scala geometrica e le differenti proprietà fisiche, il regime convettivo sviluppato all'interno della cavità è paragonabile.

Il numero di **Reynolds** dell'acqua risulta invece leggermente inferiore, poiché la maggiore viscosità cinematica tende a smorzare i moti convettivi; tuttavia, l'ordine di grandezza resta confrontabile con quello dell'aria.

Se l'acqua fosse stata analizzata in una cavità di dimensioni maggiori ( $L = 10$  cm), i valori di Rayleigh e Reynolds sarebbero cresciuti sensibilmente, portando il flusso in una condizione di **transizione verso il regime turbolento**.

La scelta di una geometria ridotta ( $L = 2$  cm) permette invece di mantenere un comportamento **stabile e laminare**, più adatto alla validazione numerica e al confronto diretto con il caso dell'aria.

Dal punto di vista della tesi, questa comparazione è particolarmente utile poiché dimostra come la corretta combinazione tra **scala geometrica** e **proprietà del fluido** consenta di ottenere regimi dinamici simili, mantenendo la simulazione in condizioni di stabilità numerica e di validità dell'approssimazione di Boussinesq.

Dall'analisi qualitativa dei campi di temperatura e velocità si possono notare differenze significative tra i due fluidi.

Nel caso della **cavità riempita d'aria** (Figura 4.7), la convezione naturale si instaura in tempi relativamente brevi: dopo pochi secondi si formano delle *celle convettive* ben definite, caratterizzate da moti ascendenti lungo la parete calda e discendenti lungo quella fredda.

La bassa viscosità e la ridotta capacità termica dell'aria favoriscono un rapido scambio termico e la formazione di vortici stabili, che tendono a occupare progressivamente l'intero dominio.

Nel caso della **cavità riempita d'acqua** (Figura 4.8), l'evoluzione risulta invece molto più lenta.

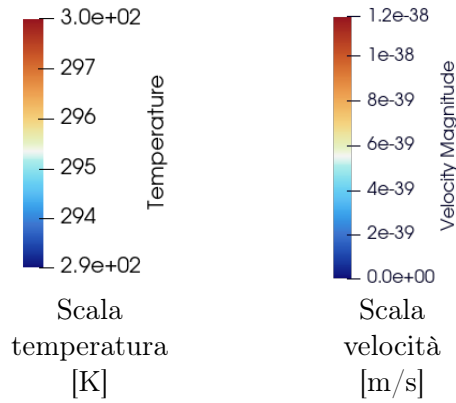
La maggiore densità, viscosità e capacità termica dell'acqua smorzano la convezione, rendendo i moti fluidi più deboli e meno turbolenti.

Nei primi istanti il trasporto avviene prevalentemente per conduzione, mentre solo dopo alcune decine di secondi iniziano a comparire deboli celle convettive nella parte superiore della cavità.

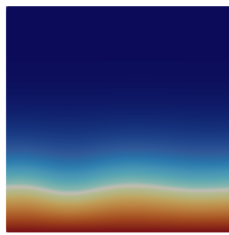
Con il progredire del tempo, il campo di temperatura tende a stabilizzarsi, evidenziando un gradiente termico più uniforme rispetto al caso dell'aria.

In sintesi, le simulazioni confermano come le proprietà fisiche del fluido (numero di Prandtl e di Rayleigh) influenzino fortemente la dinamica della convezione naturale: l'aria, con un moto più rapido, rappresenta un regime a convezione sviluppata, mentre l'acqua manifesta una risposta più lenta e diffusa, dominata inizialmente dalla conduzione termica.

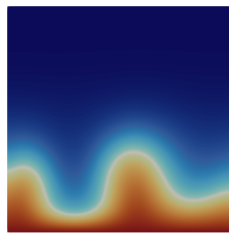
## Cavità riempita d'aria



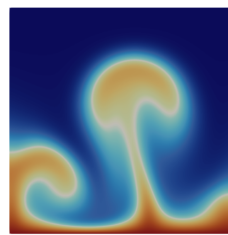
Temperatura al tempo 6 s



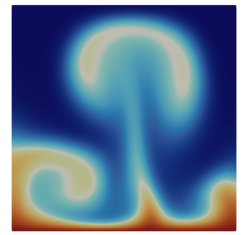
Temperatura al tempo 17 s



Temperatura al tempo 19 s



Temperatura al tempo 20,8 s



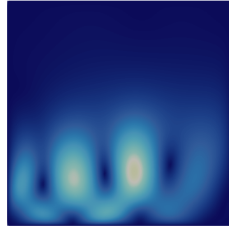
Temperatura al tempo 21.5 s



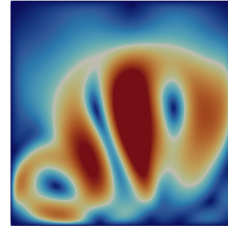
Velocità al tempo 6 s



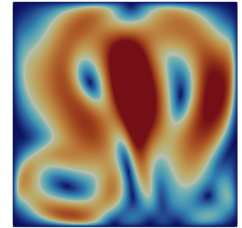
Velocità al tempo 17 s



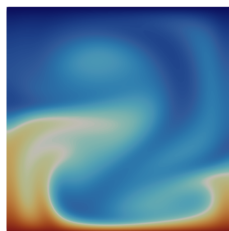
Velocità al tempo 19 s



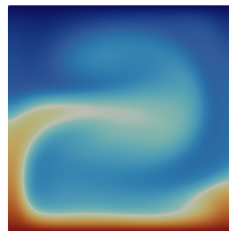
Velocità al tempo 20,8 s



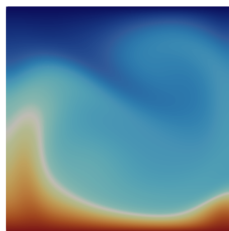
Velocità al tempo 21.5 s



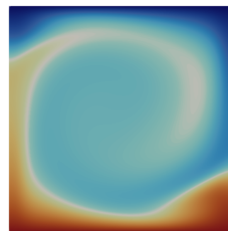
Temperatura al tempo 24 s



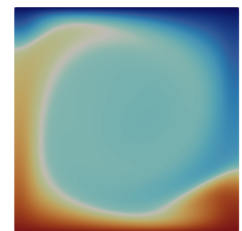
Temperatura al tempo 26 s



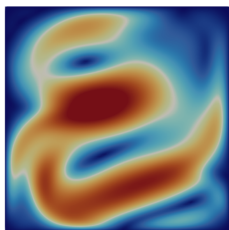
Temperatura al tempo 28 s



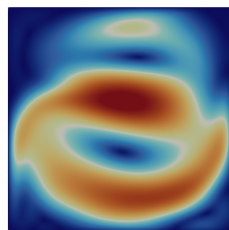
Temperatura al tempo 40 s



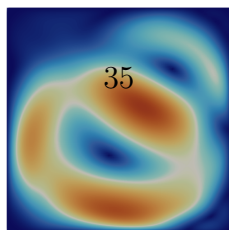
Temperatura al tempo 50 s



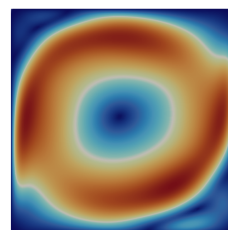
Velocità al tempo 24 s



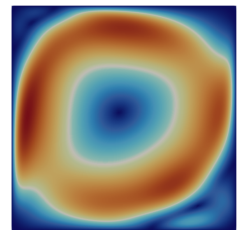
Velocità al tempo 26 s



Velocità al tempo 28 s



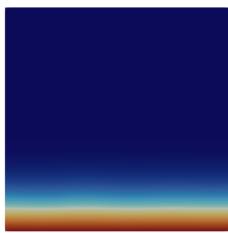
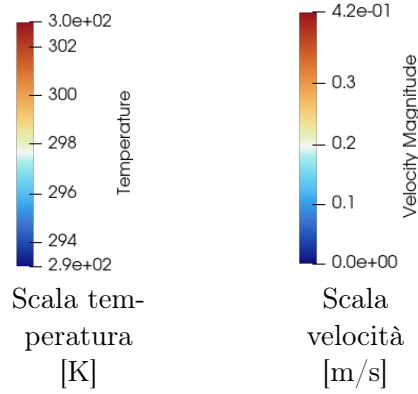
Velocità al tempo 40 s



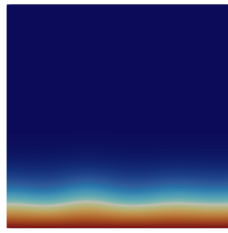
Velocità al tempo 50 s

Figura 4.7: Evoluzione temporale della cavità riempita d'aria: le righe superiori mostrano la temperatura [K], mentre quelle inferiori rappresentano il campo di velocità [m s<sup>-1</sup>] corrispondente.

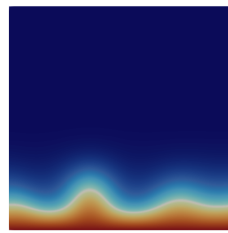
### Cavità riempita d'acqua



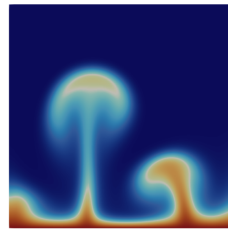
Temperatura al tempo 27 s



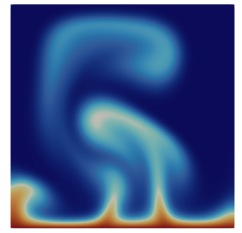
Temperatura al tempo 30 s



Temperatura al tempo 32.9 s



Temperatura al tempo 38.2 s



Temperatura al tempo 41.5 s



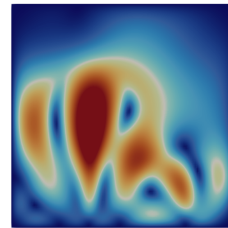
Velocità al tempo 27 s



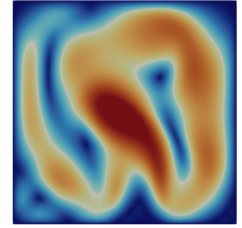
Velocità al tempo 30 s



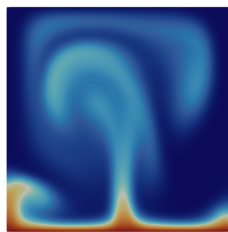
Velocità al tempo 32.9 s



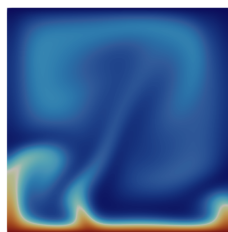
Velocità al tempo 38.2 s



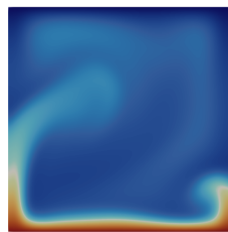
Velocità al tempo 41.5 s



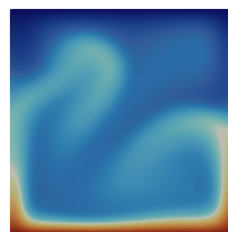
Temperatura al tempo 44 s



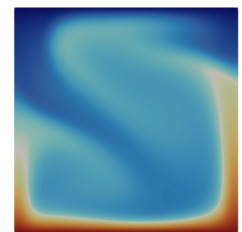
Temperatura al tempo 49 s



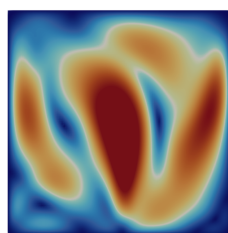
Temperatura al tempo 55 s



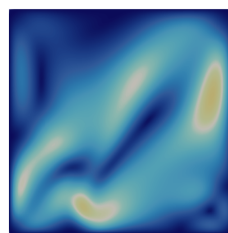
Temperatura al tempo 77 s



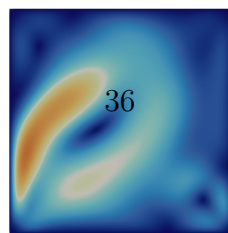
Temperatura al tempo 100 s



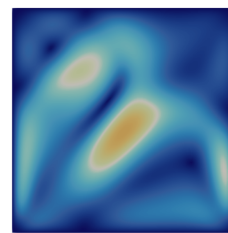
Velocità al tempo 44 s



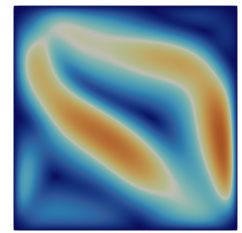
Velocità al tempo 49 s



Velocità al tempo 55 s



Velocità al tempo 77 s



Velocità al tempo 100 s

Figura 4.8: Evoluzione temporale della cavità riempita d'acqua: le righe superiori mostrano la temperatura [K], mentre quelle inferiori rappresentano il campo di velocità [ $\text{m s}^{-1}$ ] corrispondente.

# Capitolo 5

## Conclusioni

Il lavoro di tesi ha permesso di sviluppare un percorso metodologico completo per la simulazione numerica di fenomeni termo-fluidodinamici, utilizzando esclusivamente strumenti open-source di elevata versatilità e potenza computazionale.

Attraverso l'utilizzo combinato di **Gmsh**, **FEniCSx** e **ParaView**, è stato possibile realizzare un flusso di lavoro integrato comprendente tutte le fasi della simulazione: dalla costruzione della geometria e della mesh, alla formulazione variazionale del problema, fino alla visualizzazione e all'analisi qualitativa dei risultati.

La progressione dei casi studiati ha consentito di comprendere in modo graduale i principi fisici alla base della conduzione e della convezione termica.

Il problema stazionario di conduzione ha fornito un punto di partenza per la verifica della correttezza numerica del modello, mentre l'introduzione della dipendenza temporale ha permesso di analizzare l'evoluzione dinamica del campo termico e la stabilizzazione del sistema verso il regime permanente.

Il flusso laminare di Poiseuille ha consentito di validare il comportamento idrodinamico del codice, confermando la linearità della distribuzione di velocità tipica di un regime laminare.

Infine, la cavità riscaldata ha rappresentato il banco di prova più completo, poiché in essa si manifesta l'interazione tra campi termici e fluidodinamici. Il confronto tra aria e acqua ha evidenziato come la diversa combinazione di densità, viscosità e diffusività termica influenzi fortemente l'intensità dei moti convettivi e la distribuzione delle isoterme.

L'analisi dei numeri di Reynolds e Rayleigh ha confermato la coerenza fisica dei risultati e la corretta previsione del regime laminare nelle condizioni considerate.



Nel complesso, il lavoro ha dimostrato la capacità delle metodologie numeriche implementate di fornire risultati affidabili e riproducibili, ponendo le basi per studi futuri in ambiti più complessi.

In prospettiva, la metodologia sviluppata potrà essere estesa a sistemi tridimensionali o a casi con accoppiamento multiphysics, come lo studio dello *streaming acustico* o l'impiego di materiali a cambiamento di fase (PCM).

Tali sviluppi permetterebbero di esplorare con maggiore dettaglio l'efficienza dei processi di scambio termico e di ottimizzare la progettazione di sistemi energetici avanzati.

## Repository GitHub del progetto

Per garantire la riproducibilità delle simulazioni numeriche presentate, è stato realizzato un repository pubblico su **GitHub**, che raccoglie il materiale di sviluppo e di supporto utilizzato nel corso dell'elaborato.

Nel repository sono contenuti:

- i codici sorgente in **Python**
- una copia del tutorial riportato in Appendice, che illustra passo-passo la procedura di configurazione e di esecuzione delle simulazioni;

Il repository ha lo scopo di fornire una base di partenza per l'analisi termo-fluidodinamica numerica di casi simili, e di agevolare la comprensione del workflow proposto, che integra **Gmsh**, **FEniCSx** e **ParaView** in un'unica catena di calcolo open-source.

Tutto il materiale è disponibile pubblicamente al seguente indirizzo:

<https://github.com/FrancescaManzoni/Simulazioni-Termo-Fluidodinamiche-con-FEniCSx>

# Appendice A

## Appendice

### A.1 Introduzione

In questo tutorial mostriamo come realizzare una simulazione termo-fluidodinamica utilizzando strumenti open-source: **Gmsh** per la generazione della mesh, **Docker** per l'ambiente di calcolo, **FEniCSx** per la risoluzione numerica e **ParaView** per la visualizzazione dei risultati.

#### A.1.1 Prerequisiti

Per seguire correttamente il tutorial è necessario disporre dei seguenti strumenti e file:

- un file `.geo` di Gmsh, da cui generare la mesh del dominio;
- il file `.msh` esportato da Gmsh, contenente la discretizzazione e i tag di contorno;
- un file `.py` scritto in Python con FEniCSx, che implementa la simulazione numerica;
- un **Dockerfile**, che definisce l'ambiente software necessario per eseguire FEniCSx;
- **Docker Desktop** installato e avviato, per gestire i container ed eseguire lo script Python;
- **ParaView**, per aprire i file di output (`.xdmf`) e visualizzare i risultati.

Si raccomanda di mantenere tutti i file del progetto nella **stessa cartella** per semplificare la gestione e l'esecuzione del tutorial.

## A.2 Caso di studio

Il caso di studio scelto come esempio applicativo del metodo di simulazione numerica riguarda una **cavità quadrata bidimensionale** di lato  $H = 0.10$  m, completamente chiusa e riempita di acqua. Si tratta di una configurazione classica per l'analisi termo-fluidodinamica, che consente di verificare l'efficienza del modello nel risolvere un problema accoppiato di moto e trasporto termico.

La mesh del dominio, generata in `Gmsh` e salvata come `geometria.msh`, è strutturata e composta da celle quadrate di dimensioni uniformi. La simulazione è di tipo **transiente** e **incomprimibile**, condotta nel dominio del tempo a partire da condizioni iniziali di quiete e temperatura uniforme.

### A.2.1 Condizioni al contorno

Le condizioni al contorno applicate sono le seguenti:

- **Parete inferiore:** mantenuta a temperatura costante pari a  $T_{\text{bottom}} = 298$  K;
- **Parete superiore:** mantenuta a temperatura costante pari a  $T_{\text{top}} = 293$  K;
- **Pareti laterali:** adiabatiche, ovvero con flusso termico nullo;
- **Tutte le pareti:** condizione di no-slip, cioè velocità nulla in ogni punto del contorno.

Il campo di temperatura iniziale è uniforme e pari a  $T_0 = 293$  K, mentre la velocità iniziale è nulla in tutto il dominio.

Il fluido considerato è aria ( $T_{\text{ref}} = 293$  K). I valori adottati per le proprietà fisiche sono riportati nella Tabella 4.1.

## A.3 Creazione della mesh su Gmsh

La geometria del dominio è costruita mediante il software `Gmsh`. Si tratta di una cavità quadrata di lato 0.1 m, discretizzata con una mesh strutturata composta da celle quadrate regolari.

Il file di definizione della geometria, scritto in linguaggio `.geo`, è riportato di seguito:

```

1 // Quadrato 0.1 x 0.1 con celle QUADRATE n x n
2 SetFactory("OpenCASCADE");
3
4 // Parametri
5 L = 0.1;
6 H = 0.1;
7 n = 40; // stesse divisioni su x e y -> lato cella = 0.1/n
8
9 // Geometria
10 Point(1) = {0, 0, 0, 1};
11 Point(2) = {0, H, 0, 1};
12 Point(3) = {L, H, 0, 1};
13 Point(4) = {L, 0, 0, 1};
14
15 Line(1) = {1, 2}; // x=0
16 Line(2) = {2, 3}; // y=H
17 Line(3) = {3, 4}; // x=L
18 Line(4) = {4, 1}; // y=0
19
20 Curve Loop(1) = {1, 2, 3, 4};
21 Plane Surface(1) = {1};
22
23 // Gruppi fisici (opzionali)
24 Physical Curve("left", 10) = {1};
25 Physical Curve("top", 11) = {2};
26 Physical Curve("right", 12) = {3};
27 Physical Curve("bottom", 13) = {4};
28 Physical Surface("fluid", 14) = {1};
29
30 // Mesh strutturata quadrata
31 Transfinite Curve {2, 4} = n Using Progression 1; // orizzontali (lungo x)
32 Transfinite Curve {1, 3} = n Using Progression 1; // verticali (lungo y)
33 Transfinite Surface {1} = {1, 2, 3, 4};

```

Questo script definisce:

- i quattro vertici del quadrato mediante comandi `Point`;
- le quattro linee di contorno con i comandi `Line`;

- la superficie piana interna, delimitata dal **Curve Loop**;
- i **Physical Groups**, utilizzati per assegnare un nome logico alle pareti (utile per le condizioni al contorno);
- la mesh strutturata, generata tramite il comando **Transfinite**, che garantisce una suddivisione regolare in celle quadrate di dimensioni uniformi.

La mesh così definita deve essere esportata direttamente dal software **Gmsh** in formato **.msh** (versione 2) e salvata con il nome **geometria.msh**, in modo da poter essere successivamente importata dallo script **Python** per la risoluzione numerica del problema.

## A.4 Creazione del DockerFile

Per garantire un ambiente riproducibile utilizziamo un **Dockerfile**, ovvero un file di testo *senza estensione* (il nome deve essere esattamente **Dockerfile**) posto nella stessa cartella del progetto

```
1 FROM dolfinx/dolfinx:stable
2 RUN python3 -m pip install --no-cache-dir meshio gmsh
3 WORKDIR /workspace
4 CMD ["bash"]
```

Spiegazione delle direttive:

- **FROM dolfinx/dolfinx:stable**: immagine base ufficiale con FEniCSx/DOLFINx preinstallato .
- **RUN python3 -m pip install ...**: installa i pacchetti Python aggiuntivi necessari al tutorial: **meshio** (lettura/scrittura mesh) e **gmsh** (API Python).
- **WORKDIR /workspace**: imposta la cartella di lavoro interna al container dove monteremo i file del progetto.
- **CMD ["bash"]**: all'avvio del container apre una shell interattiva; da qui potremo lanciare **python3**.

## A.5 Docker Desktop

Per eseguire i container creati con il `Dockerfile` è necessario che **Docker Desktop** sia installato e in esecuzione.

Docker Desktop fornisce l'interfaccia grafica che gestisce i container e i volumi sul sistema operativo (Windows o macOS), permettendo di avviare, monitorare ed eliminare i container in modo semplice.

Dopo l'installazione, è sufficiente aprire Docker Desktop e verificare che l'icona nella barra delle applicazioni indichi lo stato "Running".

Se Docker Desktop non è attivo, i comandi `docker build` e `docker run` non saranno disponibili.

Nel nostro caso Docker Desktop rimane aperto in background mentre, dal terminale vengono eseguiti i comandi per costruire l'immagine e lanciare il container. Una volta avviato, Docker Desktop permette di visualizzare:

- le immagini disponibili (es. `cavita-fenicsx` create dal nostro `Dockerfile`);
- i container in esecuzione (dove lanciato lo script Python);

È importante mantenere Docker Desktop sempre attivo durante le fasi di compilazione (`docker build`) ed esecuzione (`docker run`). In questo modo l'ambiente predisposto con il `Dockerfile` sarà pienamente operativo e lo script `cavita.py` potrà essere eseguito senza problemi.

## A.6 File Python

Il file Python costituisce la parte centrale del progetto, nella quale viene implementata la simulazione numerica.

In questo script vengono importati i pacchetti necessari, letta la mesh generata in `Gmsh`, definite le costanti del problema, costruiti gli spazi funzionali per le variabili incognite e risolto, passo dopo passo, il sistema di equazioni differenziali che governa il moto e il trasporto termico del fluido. ù

La struttura dello script segue un ordine logico suddiviso in blocchi funzionali:

1. Importazione delle librerie;
2. Importazione della mesh da **Gmsh**;
3. Definizione delle costanti e dei parametri di simulazione;
4. Costruzione degli spazi funzionali;
5. Impostazione delle condizioni iniziali e al contorno;
6. Formulazione variazionale del problema;
7. Integrazione temporale;
8. Esportazione dei risultati.

Nei paragrafi seguenti ciascun passaggio è descritto nel dettaglio.

### A.6.1 Importazione delle librerie

```
1 import os
2 import numpy as np
3 from mpi4py import MPI
4 from petsc4py import PETSc
5 import ufl
6
7 from dolfinx import mesh as dmesh, fem, io
8 from dolfinx.io import gmshio
9 from dolfinx.fem.petsc import LinearProblem
10 from basix.ufl import element, mixed_element
11
12 comm = MPI.COMM_WORLD
13 rank = comm.rank
14 TAG = (lambda s: print(s, flush=True)) if rank == 0 else (lambda s: None)
```

Questa sezione prepara l'ambiente di lavoro importando tutte le librerie necessarie.

- **NumPy** gestisce le operazioni numeriche elementari, in particolare le manipolazioni di vettori e matrici.

- **mpi4py** e **petsc4py** forniscono il supporto alla parallelizzazione, rendendo il codice compatibile con architetture multi-processore (MPI).
- **UFL** (Unified Form Language) consente di scrivere in modo simbolico le equazioni nella forma variazionale, indipendentemente dal tipo di elemento finito utilizzato.
- **DOLFINx** costituisce il motore principale di FEniCSx: gestisce la mesh, gli spazi funzionali, la costruzione dei termini integrali e la risoluzione dei sistemi lineari.
- **Basix** definisce la natura e il grado degli elementi finiti impiegati nella discretizzazione.

## A.6.2 Importazione della mesh da Gmsh

```

1 msh_name = "geometria.msh"
2 if rank == 0 and not os.path.exists(msh_name):
3     raise FileNotFoundError(msh_name)
4 ret = gmshio.read_from_msh(msh_name, comm, gdim=2)
5 mesh = ret if isinstance(ret, dmsh.Mesh) else ret[0]
6
7 x = mesh.geometry.x
8 xmin, ymin = float(np.min(x[:,0])), float(np.min(x[:,1]))
9 xmax, ymax = float(np.max(x[:,0])), float(np.max(x[:,1]))
10 H = max(xmax-xmin, ymax-ymin)

```

In questo blocco il codice importa la mesh, precedentemente generata in **Gmsh** e salvata come `geometria.msh`.

Il file viene letto con la funzione `gmshio.read_from_msh`, che restituisce un oggetto `mesh` contenente le informazioni geometriche e topologiche del dominio.

La dimensione geometrica `gdim=2` indica che il problema è bidimensionale.

## A.6.3 Definizione delle costanti e dei parametri numerici

```

1 rho0 = 1.225
2 mu   = 1.8e-5
3 nu   = mu / rho0           # 1.47e-5 m^2/s
4 k    = 0.025

```



```

5 cp      = 1005.0
6 alpha = k / (rho0 * cp)          #      2.03e-5 m^2/s
7 beta  = 3.0e-3
8 g      = 9.81
9 g_vec  = ufl.as_vector((0.0, -g))
10 t_end  = 100.0
11 dt     = 0.01
12 nsteps = int(round(t_end/dt))
13 save_every = 10

```

Qui vengono definiti i parametri che caratterizzano il fluido e il calcolo numerico.

Le costanti fisiche ( $\rho_0, \mu, k, c_p, \beta, g$ ) sono proprietà dell'acqua a temperatura ambiente. Da queste si derivano due grandezze adimensionali di rilievo numerico:  $\nu$ , la viscosità cinematica, e  $\alpha$ , la diffusività termica.

Segue la definizione dei parametri temporali della simulazione: `t_end` rappresenta la durata complessiva (100s), `dt` è il passo temporale (0.01s), e `nsteps` il numero totale di iterazioni temporali.

Il parametro `save_every` controlla la frequenza di salvataggio dei risultati, in modo da ridurre il numero di file scritti senza compromettere la risoluzione temporale dell'analisi.

#### A.6.4 Costruzione degli spazi funzionali

```

1 cell = mesh.ufl_cell().cellname()
2 Ve = element("Lagrange", cell, 2, shape=(2,))
3 Qe = element("Lagrange", cell, 1)
4 Te = element("Lagrange", cell, 2)
5 W = fem.functionspace(mesh, mixed_element([Ve, Qe]))
6 V, _ = W.sub(0).collapse()
7 Q, _ = W.sub(1).collapse()
8 Th = fem.functionspace(mesh, Te)

```

Gli spazi funzionali definiscono la base numerica su cui vengono proiettate le variabili incognite. La funzione `element` stabilisce il tipo di elemento finito associato a ciascuna grandezza:

- **Velocità (Ve):** elementi di tipo Lagrange di grado 2 con due componenti, corrispondenti alle direzioni  $x$  e  $y$ ;

- **Pressione** ( $Q_e$ ): elementi Lagrange di grado 1;
- **Temperatura** ( $T_e$ ): elementi Lagrange di grado 2.

La combinazione tra elementi quadratici per la velocità e lineari per la pressione forma la cosiddetta coppia di **Taylor–Hood**, che assicura stabilità numerica per la discretizzazione delle equazioni di Navier–Stokes incomprimibili.

La variabile  $W$  rappresenta lo spazio funzionale misto composto dalle due variabili  $(\mathbf{u}, p)$ . Le funzioni  $V$  e  $Q$  vengono poi ricavate come sottospazi di  $W$ , mentre  $Th$  è lo spazio dedicato alla temperatura.

Questa organizzazione permette di risolvere contemporaneamente il campo di velocità e pressione in un unico sistema lineare accoppiato, e di trattare la temperatura in un problema separato ma collegato.

### A.6.5 Condizioni iniziali e al contorno

```

1 u_zero = fem.Function(V); u_zero.x.array[:] = 0.0
2 T_bottom = 298.0
3 T_top    = 293.0
4 T_init   = 293.0
5 T_ref    = 293.0

```

In questo blocco vengono impostate le condizioni di partenza della simulazione. Il vettore di velocità iniziale ( $\mathbf{u\_zero}$ ) è nullo in tutto il dominio, mentre la temperatura iniziale è uniforme e pari alla temperatura di riferimento ( $T_{\text{ref}} = 293 \text{ K}$ ).

Le condizioni al contorno sono di tipo Dirichlet per le pareti isoterme (superiore e inferiore) e di tipo Neumann omogenee (naturali) per le pareti laterali, che sono adiabatiche.

La condizione di *no-slip* per la velocità impone che ogni punto del contorno sia solidale con la parete, annullando la velocità tangenziale e normale.

### A.6.6 Formulazione variazionale

```

1 a_mom = (
2     (1.0/dt)*ufl.inner(U, v)*dx
3     + ufl.inner(ufl.grad(U)*u_n, v)*dx
4     + 2*nu*ufl.inner(ufl.sym(ufl.grad(U)), ufl.sym(ufl.grad(v)))*dx
5     - ufl.inner(P, ufl.div(v))*dx
6     + ufl.inner(ufl.div(U), q)*dx
7 )
8 L_mom = (1.0/dt)*ufl.inner(u_n, v)*dx - beta*ufl.inner((Tn - T_ref)*g_vec,
    v)*dx

```

Le equazioni differenziali vengono espresse nella loro forma variazionale (o debole), ossia moltiplicate per una funzione di test e integrate sul dominio.

Questo approccio è alla base del metodo degli elementi finiti e consente di gestire geometrie complesse e condizioni al contorno arbitrarie.

La prima forma (`a_mom`) rappresenta i termini del bilancio della quantità di moto, mentre la seconda (`L_mom`) costituisce il termine noto del sistema.

Il termine temporale  $(1/dt) \text{inner}(U, v)$  impone la dipendenza nel tempo, mentre i termini con il gradiente di  $\mathbf{u}$  e la viscosità  $\nu$  rappresentano la diffusione della quantità di moto.

Il termine  $-\beta(T - T_{\text{ref}}) \mathbf{g}$  introduce la dipendenza della densità dalla temperatura.

## A.6.7 Integrazione temporale

```

1 for n in range(1, nsteps+1):
2     t = n*dt
3     w_sol = problem_NS.solve()
4     Uh, map_u = W.sub(0).collapse()
5     Ph, map_p = W.sub(1).collapse()
6     u_out = fem.Function(Uh); u_out.x.array[:] = w_sol.x.array[map_u]
7     p_out = fem.Function(Ph); p_out.x.array[:] = w_sol.x.array[map_p]
8
9     T_out = problem_T.solve()
10
11     u_n.interpolate(u_out)
12     p_n.interpolate(p_out)
13     Tn.x.array[:] = T_out.x.array

```

Il ciclo temporale rappresenta il cuore della simulazione transiente. Ad ogni iterazione:

1. vengono risolti i campi di velocità e pressione (`problem_NS`);
2. viene risolta l'equazione della temperatura (`problem_T`);
3. i risultati vengono salvati e utilizzati come condizioni iniziali per il passo successivo.

Il tempo simulato avanza di un intervallo  $\Delta t$  a ogni ciclo.

L'algoritmo implementa quindi una soluzione temporale esplicita del sistema accoppiato, aggiornando le variabili primarie ( $\mathbf{u}, p, T$ ) fino al tempo finale.

### A.6.8 Esportazione dei risultati

```
1 xdmf = io.XDMFFile(mesh.comm, "results_transient.xdmf", "w")
2 xdmf.write_mesh(mesh)
```

Il file di output è scritto in formato **XDMF**, compatibile con **ParaView**. Questo formato conserva sia la geometria del dominio sia i valori delle variabili ai nodi della mesh.

Al termine della simulazione il file `results_transient.xdmf` conterrà la sequenza temporale dei campi di velocità, pressione e temperatura, pronti per l'analisi e la visualizzazione.

L'intero script costituisce un esempio completo di configurazione numerica in **FEniCSx**, capace di risolvere un problema termofluidodinamico transiente con geometria arbitraria e condizioni al contorno personalizzabili.

## A.7 Esecuzione della simulazione in ambiente Docker

Una volta predisposti tutti i file del progetto (`geometria.msh`, `simulazione.py` e `Dockerfile`), è possibile eseguire la simulazione all'interno di un ambiente virtualizzato tramite **Docker**. In questo modo si garantisce la completa portabilità del codice, evitando problemi di compatibilità tra versioni di librerie o sistemi operativi.

Per avviare la simulazione è necessario utilizzare il terminale **PowerShell** di Windows (o un terminale equivalente su altri sistemi operativi) e posizionarsi esattamente nella cartella che contiene tutti i file del progetto. L'intera procedura si articola in due passaggi principali:

1. **Creazione dell'immagine Docker:** questo comando costruisce un'immagine locale basata sul Dockerfile presente nella directory corrente.

```
1 docker build -t simulazione_termofluidodinamica .
```

2. **Esecuzione del container e avvio della simulazione:** una volta creata l'immagine, la simulazione può essere lanciata tramite:

```
1 docker run --rm -v "${PWD}:/workspace" simulazione_termofluidodinamica  
python3 /workspace/simulazione.py
```

Nel dettaglio:

- `--rm` elimina automaticamente il container al termine dell'esecuzione, evitando accumulo di risorse temporanee;
- `-v "${PWD}:/workspace"` monta la cartella locale nella directory di lavoro del container, rendendo accessibili i file del progetto all'ambiente virtuale;
- `simulazione_termofluidodinamica` è il nome dell'immagine Docker creata nel passaggio precedente;
- `python3 /workspace/simulazione.py` esegue lo script Python all'interno del container.

Al termine dell'esecuzione, nella stessa cartella di lavoro verrà generato il file di output `results_transient` contenente i campi di temperatura, pressione e velocità, successivamente visualizzabili e analizzabili in ParaView.

**Apertura dei file** Dopo aver lanciato la simulazione con successo, nella cartella del progetto sono presenti i file:

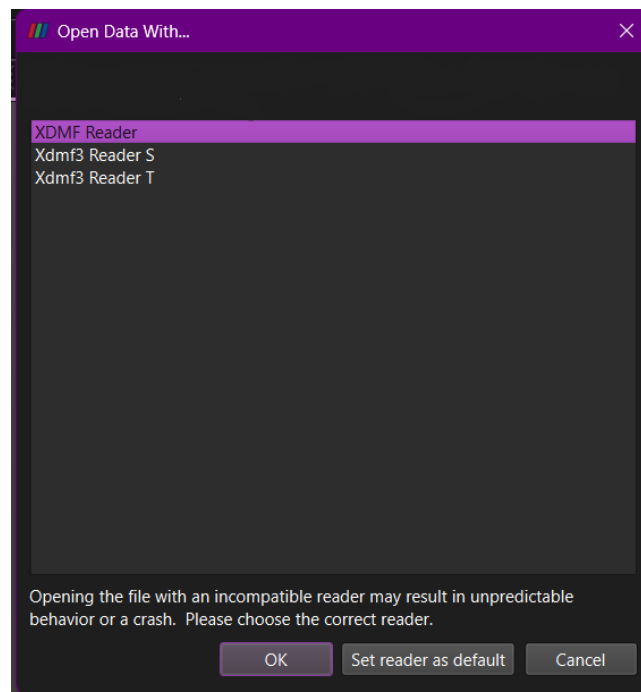
- `temperature.xdmf` — campo di temperatura,
- `velocity.xdmf` — campo di velocità,
- `pressure.xdmf` — campo di pressione.

## A.8 Visualizzazione in Paraview

In ParaView è sufficiente aprire uno di questi file e, dal menu *Pipeline Browser*, selezionare il campo da visualizzare. Gli altri file possono essere caricati successivamente per effettuare analisi comparative.

ParaView riconosce automaticamente se i file contengono una serie temporale.

- Nel caso di **simulazioni stazionarie**, è sufficiente aprire il file e visualizzare il risultato al tempo unico disponibile (esecutore **S**).
- Nel caso di **simulazioni transitorie**, ParaView offre la possibilità di scorrere tra i vari time-step (esecutore **T**), creando anche animazioni dell'evoluzione temporale.



Alcuni esempi di operazioni utili:

- colorare il dominio secondo il campo scalare di interesse (es. temperatura  $T$ );
- fissare una scala di colori coerente per confronti temporali;
- esportare immagini o animazioni in formato video.

La visualizzazione con ParaView costituisce l'ultimo passo del workflow: dopo aver definito la geometria, costruito la mesh, impostato l'ambiente con Docker, e risolto numericamente il problema con FEniCSx, i risultati diventano leggibili e interpretabili in forma grafica. Questa fase consente di analizzare i campi di pressione, velocità e temperatura

## A.9 Conclusione

L'intero processo dimostra come, partendo da strumenti open-source e modulari, sia possibile impostare e risolvere un problema di convezione termica in modo sistematico, riproducibile e indipendente dal sistema operativo.

La struttura del codice e dell'ambiente di calcolo presentati può essere facilmente adattata ad altri domini geometrici, fluidi o condizioni al contorno, costituendo così una base di riferimento per successive analisi e sviluppi numerici.

# Bibliografia

- [1] J. S. Dokken, “Fundamentals — code example,” 2023.
- [2] J. S. Dokken, “Nitsche’s method — dolfinx tutorial,” 2023.
- [3] J. S. Dokken, “Neumann and dirichlet boundary conditions — code example,” 2023.
- [4] J. S. Dokken, “Heat equation — dolfinx tutorial,” 2023.
- [5] J. S. Dokken, “Heat equation — code example,” 2023.
- [6] J. S. Dokken, “Singular poisson problem — dolfinx tutorial,” 2023.
- [7] J. S. Dokken, “Navier–stokes equations — code example,” 2023.
- [8] J. S. Dokken, “Multiple dirichlet conditions — dolfinx tutorial,” 2023.
- [9] FEniCS Project, “Fenics project community,” 2024.
- [10] C. Geuzaine and J.-F. Remacle, “Gmsh official website,” 2025.
- [11] Kitware Inc., “Paraview official website,” 2025.
- [12] Docker Inc., “Docker documentation,” 2025.
- [13] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 7th ed., 2011.
- [14] M. Rossi, “Dispense del corso di termofluidodinamica,” 2024.



- [15] A. Logg, K.-A. Mardal, and G. N. Wells, *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, vol. 84 of *Lecture Notes in Computational Science and Engineering*. Springer, 2012.
- [16] C. Geuzaine and J.-F. Remacle, *Gmsh: A Three-Dimensional Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities*. University of Liège, 2009. Software open-source distribuito con licenza GNU GPL v2 o successive.
- [17] FEniCS Project, *FEniCSx: Automated Solution of Differential Equations by the Finite Element Method*. FEniCS Project, 2024. Software open-source distribuito con licenza GNU LGPL; logo con licenza CC BY-SA 4.0.
- [18] Kitware Inc., *ParaView: Open-Source Parallel Visualization Application*. Kitware Inc., 2025. Software open-source distribuito con licenza BSD a 3 clause.