



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dipartimento di Ingegneria industriale

Corso di Laurea in
INGEGNERIA AEROSPAZIALE

Studio e implementazione CFD-CHT dello scambio termico in un alveare

Tesi di Laurea in
ING-IND/06: Fluidodinamica

Relatore

Prof. Guglielmo Minelli

Presentata da

Maurizio Ceré

Sessione Ottobre 2025

Anno Accademico 2024/2025

Sommario

Nel presente lavoro si è indagato come la sopravvivenza delle colonie di *Apis mellifera* — già oggi messa a rischio dai cambiamenti climatici e dalle crescenti instabilità ambientali — possa essere studiata in modo prevedibile mediante l'adozione di strumenti e software numerici open-source, in alternativa a un approccio esclusivamente sperimentale.

A tale scopo sono state condotte simulazioni CHT-CFD (*Conjugate Heat Transfer – Computational Fluid Dynamics*) mirate al confronto di differenti modelli geometrici, con particolare attenzione alla posizione del glomere e alla sua influenza sulla quantità di calore disperso.

Dall'analisi è emerso che gli alveari con sviluppo verticale, più vicini per configurazione ai nidi naturali, presentano una dispersione di calore sensibilmente ridotta, fino al 30% in meno rispetto alle configurazioni tradizionali generalmente adottate nell'apicoltura moderna.

Indice

1	Introduzione	1
1.1	La sfida della termoregolazione in apicoltura	2
1.2	Meccanismi di termoregolazione delle api mellifere	3
1.2.1	Termoregolazione attiva: produzione di calore	3
1.2.2	Termoregolazione passiva: isolamento e design dell'alveare . .	4
1.3	Tecnologie e soluzioni per il supporto alla termoregolazione	5
1.4	Modellazione numerica e simulazioni CFD	5
2	Conjugate Heat Transfer (CHT): Principio e applicazioni ingegneristiche	7
2.1	OpenFOAM	8
2.2	CalculiX	8
2.3	preCICE	8
3	Modellazione	11
3.1	Fluid domain	11
3.1.1	Modello di turbolenza	12
3.1.2	buoyantPimpleFoam: file necessari e BC	13
3.2	Solid domain: modelli termo-fisici, condizioni al contorno e impostazioni numeriche	14
3.2.1	Proprietà termofisiche del solido.	14
4	Coupling	19
4.1	preciceDict	19
4.2	config.yml	20
4.3	precice-config.xml	20
5	Risultati e post-processing	23
5.1	Visualizzazione dei campi di temperatura e velocità	23
5.2	Time window e stabilità numerica	24
5.3	Funzioni nel controlDict.	26
5.3.1	Analisi del flusso termico uscente da BroodHole nel tempo . .	26
5.3.2	Comparazione andamento del flusso sperimentale e teorico .	28
5.3.3	Confronto tra potenza generata dalle api e potenza termica uscente	30
	Conclusioni	35

A Governing equations	37
B Tutorial	39
Bibliografia	46

Elenco delle figure

3.1	Mesh fluido nei due casi sopracitati. In entrambi i casi il dominio di fluido è lungo 2 m e alto 1 m ed è visibile il foro nel quale verrà allocato il solido	12
3.2	Solidi quotati: in nero è visibile la parte del <i>Telaio</i> , in arancione la parte del <i>Brood</i> : le misure per quest'ultimo sono state mantenute invariate per consentire una comparazione ragionevole	15
5.1	Visualizzazione del telaio solido riscaldato trascorsi 1800 s: con NT è indicata la temperatura dei nodi del solido.	23
5.2	Campo di temperatura e linee di flusso nel caso a cassetta doppia (sopra) e cassetta singola (sotto): con T è indicata la temperatura del fluido, mentre con NT la temperatura dei nodi del solido, in bianco sono rappresentate le isoterme del fluido	25
5.3	Grafico del <i>Courant Number</i> vs <i>Time</i> nel caso a cassetta doppia (a) e singola (b)	25
5.4	Grafico del calore e della potenza vs <i>time</i> dispersi da Brood nel caso a cassetta doppia (a) e singola (b).	27
5.5	Analisi delle potenze in scala semilogaritmica e $\propto 1/\sqrt{t}$. In blu sono rappresentati i dati numerici, mentre in azzurro la curva interpolante	29
5.6	Calore disperso (rosso) e calore generato (nero) con modello a potenza costante	32
5.7	Calore disperso (rosso) e potenza generata (nero) con modello a potenza costante, i grfici non si intersecano se si osservano le scale, tuttavia è stato necessario porli in questo modo per motivi di rappresentazione	34

Capitolo 1

Introduzione

Gli alveari rappresentano sistemi biologici altamente complessi, le cui dinamiche interne sono ancora oggi oggetto di studio e in parte poco comprese. Il ruolo cruciale delle api mellifere (*Apis mellifera*) per la sopravvivenza degli ecosistemi e per la produttività agricola è essenziale ed al contempo minacciato da diversi fattori ambientali. I cambiamenti climatici costituiscono, la sfida principale esponendo le colonie a condizioni estreme, con un conseguente aumento dei fattori di rischio causandone sempre più spesso il collasso.

In questo contesto i rigidi inverni costituiscono una delle condizioni maggiormente gravose: durante questo periodo le colonie sono esposte a basse temperature, umidità elevata e limitata disponibilità di risorse. In tali circostanze si rende fondamentale il mantenimento del nido sopra i 10 °C (con una temperatura media di 21°C in assenza della covata), infatti, al di sotto di questa soglia le api entrano in coma da freddo (*chill-coma*) e cessano le attività vitali [1]. Per sopperire e contrastare le perdite di calore (e analogamente per lo smaltimento di calore in condizioni di caldo eccessivo), le api operano una serie di meccanismi di termoregolazione. Sebbene la comprensione di tali meccanismi non sia totalmente chiara una serie di studi sperimentali e CFD (*Computational Fluid Dynamics*) hanno indagato questi fenomeni ed i tassi di mortalità. L'approccio numerico ha consentito di comprendere, almeno in parte, le dinamiche interne alla colonia, offrendo la possibilità di individuare e valutare eventuali soluzioni ingegneristiche senza ricorrere necessariamente a un approccio sperimentale, più oneroso e complesso da realizzare [2].

In questo contesto, il presente lavoro si propone di analizzare lo scambio termico in un alveare sottoposto a condizioni ambientali invernali critiche (temperatura esterna pari a 5 °C), attraverso lo sviluppo di un modello bidimensionale semplificato e l'impiego di simulazioni numeriche CHT-CFD (*Conjugate Heat Transfer – Computational Fluid Dynamics*).

In particolare, il lavoro si concentra sull'analisi dell'influenza dell'altezza del glomere sulla quantità di calore necessaria alla sopravvivenza della colonia, attraverso la simulazione di due differenti configurazioni: una cassetta singola e una cassetta doppia. Il percorso della tesi si articola in una prima parte introduttiva, dedicata alla descrizione del problema e alla rassegna della letteratura di riferimento; prosegue con una sezione metodologica dedicata al *Conjugate Heat Transfer* e ai software open-source utilizzati per le simulazioni; si concentra quindi sulla modellazione del

caso studio, sia per quanto riguarda il dominio solido che quello fluido, e si conclude con l'analisi dei risultati ottenuti.

1.1 La sfida della termoregolazione in apicoltura

Le api rappresentano un modello straordinario di adattamento ecologico, tuttavia il cambiamento climatico costituisce un fattore di rischio per la sopravvivenza delle colonie.

In risposta alla modifica delle condizioni circostanti le colonie hanno sviluppato un complesso ed efficiente sistema di termoregolazione collettiva [2, 3]. Questa capacità è fondamentale ai fini del mantenimento delle condizioni microclimatiche ottimali. La condizione più gravosa si verifica in presenza della covata, la cui temperatura deve essere mantenuta costantemente tra i 32 e i 36 °C per garantire lo sviluppo della prole, comportando, specialmente in condizioni invernali, la presenza di gradienti termici importanti (tra interno ed esterno) ed un enorme consumo metabolico.

In natura le colonie prediligono nidificare all'interno di cavità d'albero alte, strette e con pareti spesse, solitamente rastremate verso l'alto e con ingresso situato in basso [2]. Queste cavità offrono un isolamento termico elevato, limitano la convezione dell'aria e riducono la dispersione di calore, creando un ambiente favorevole alla termoregolazione passiva. Inoltre, la struttura verticale permette alle api una maggiore libertà di movimento per adattare la posizione della covata in base alle condizioni ambientali [2].

L'apicoltura moderna, tuttavia, si basa quasi esclusivamente su arnie artificiali standardizzate (Langstroth, Dadant Blatt, British National ecc.) caratterizzate da pareti sottili (13–19 mm), geometrie squadrate e limitata altezza verticale (200/230 mm). Tali arnie sono state progettate principalmente per la comodità di gestione da parte dell'apicoltore, piuttosto che per il benessere termico della colonia [2, 1]. La forzatura delle api in ambienti così stretti e strutturati (senza possibilità di organizzazione degli spazi da parte della colonia) hanno impedito un adattamento efficace generando perdite termiche fino al +70% rispetto ai nidi in albero, [2]. Sarà oggetto dello studio in questione verificare, seppur in un caso semplificato, come la differenza di altezza del glomere influisca sulla quantità di calore necessaria per il mantenimento delle condizioni ideali. Tramite simulazioni CFD si è confrontato il calore disperso in cavità naturali e artificiali. Queste differenze strutturali hanno rivelato un bilancio energetico molto più sfavorevole per le colonie ospitate in arnie artificiali (basse). Per mantenere la stessa temperatura del nido, esse devono impiegare una densità di api almeno 1,5 volte superiore rispetto a quanto richiesto in una cavità naturale [2, 5]. L'aumentata presenza di api, la necessità di generare quantità maggiori di calore e l'assenza di isolamento nelle arnie artificiali comporta un aumento dello stress metabolico conducendo ad un eccessivo consumo di risorse [2, 4, 1]. Queste problematiche si aggravano nei climi più rigidi, dove le basse temperature prolungate compromettono ulteriormente la termoregolazione della colonia. Le api, costrette a ricorrere a strategie termiche attive (*shivering thermogenesis*), consumano enormi quantità di energia, con il conseguente rischio di esaurire le scorte e causare il collasso dell'intera colonia [4, 6].

In sintesi, gli attuali modelli utilizzati sembrano disallineati rispetto alle esigenze termico-biologiche delle api, suggerendo la necessità di una riconsiderazione del design

delle arnie (materiali, struttura ecc.). Le differenze osservate nei comportamenti, nelle performance termiche e nei tassi di sopravvivenza tra i nidi naturali e quelli artificiali mettono in luce il valore di ricerche che riportino l'ambiente dell'alveare quanto più vicino possibile alla sua forma originaria [2, 7, 1].

1.2 Meccanismi di termoregolazione delle api mellifere

Le api mellifere (*Apis mellifera*) adottano una sofisticata strategia collettiva di termoregolazione per mantenere la temperatura del nido entro limiti compatibili con la sopravvivenza della colonia e lo sviluppo della covata (qualora presente). Questi meccanismi si distinguono in termoregolazione attiva e passiva, e si basano su una combinazione di fattori sociali, fisiologici e strutturali [8].

1.2.1 Termoregolazione attiva: produzione di calore

La termoregolazione attiva si basa sulla produzione individuale di calore da parte di api adulte, conosciute come “*api riscaldatrici*”. Queste api generano calore tramite contrazioni isometriche dei muscoli del volo (*shivering thermogenesis*), senza volare né muoversi, ma aumentando significativamente la propria temperatura corporea. Modellazioni numeriche hanno mostrato che, in tali condizioni, una singola ape può produrre fino a 500 W/kg di calore, un valore comparabile a quello generato durante il volo attivo [9].

Le api riscaldatrici possono posizionarsi all'interno di una cella vuota o sopra celle contenenti covata, elevando la temperatura locale da circa 32 °C a oltre 41 °C, con effetti termici che si estendono fino a 3–4 celle circostanti [9].

Durante l'inverno, le api adottano una strategia collettiva fondamentale: la formazione di un *cluster* compatto, al cui centro la temperatura è mantenuta attorno ai 21 °C, mentre ai margini può scendere anche sotto i 10 °C [6]. Tuttavia, esistono soglie termiche critiche che regolano l'attività metabolica del gruppo: se la temperatura centrale del cluster scende sotto i 15 °C o la periferia sotto i 5 °C, le api rispondono aumentando notevolmente il metabolismo per preservare e ripristinare condizioni compatibili con la sopravvivenza [6].

Modellazioni numeriche supportano queste osservazioni evidenziando inoltre che la presenza simultanea di più api riscaldatrici vicino alla covata può aumentare significativamente la velocità di riscaldamento: si osservano incrementi termici locali di 0.1–0.5 °C al minuto, a seconda che vi sia una o cinque api impegnate nel riscaldamento [9]. Questo comportamento termoregolatorio ha, tuttavia, un costo metabolico notevole e pertanto non può essere protratto per lunghi periodi.

È stato inoltre osservato che api isolate o in piccoli gruppi consumano fino a 17 volte più energia rispetto a quelle riunite in un cluster coeso, dove la dissipazione del calore è minimizzata [6] (solitamente poste al centro del cluster per minimizzare dispersioni termiche). Questo sottolinea l'importanza della struttura sociale ai fini della sopravvivenza collettiva, specialmente nei mesi più rigidi.

Durante gli inverni severi, infatti, l'incapacità di mantenere una temperatura interna adeguata può compromettere la sopravvivenza della colonia. Le perdite invernali possono raggiungere anche il 50% delle colonie in alcune regioni (Europa e

Nord America) [1], specialmente quando si verificano condizioni di stress termico prolungato, carenti scorte alimentari e scarso isolamento termico dell'alveare.

I meccanismi di termoregolazione attiva costituiscono un mezzo di contrasto alle difficili condizioni climatiche tanto efficaci quanto oneroso, perciò ove possibile si preferiscono attuare strategie a "costo zero", ovvero passive.

1.2.2 Termoregolazione passiva: isolamento e design dell'alveare

Oltre alla produzione attiva di calore, le api mellifere adottano strategie passive di termoregolazione per limitare la dispersione termica e non gravare metabolicamente sulla colonia, specialmente durante l'inverno. Questi comportamenti di fatto non comportano un aumento del metabolismo ma si basano su modifiche strutturali dell'ambiente del nido e sulla sua interazione con i materiali e le relative proprietà termo-fisiche.

Uno dei principali strumenti passivi è l'uso del propoli, una resina vegetale con bassa permeabilità al vapore acqueo, che le api utilizzano per sigillare fessure e piccole aperture della cavità del nido, riducendo così le perdite di calore per convezione e ventilazione forzata [7]. Questa pratica, oltre a migliorare l'isolamento termico, consente anche un migliore controllo dell'umidità interna, essenziale per la sopravvivenza invernale.

Anche la distribuzione strategica dei materiali nelle varie celle all'interno del favo contribuisce alla generazione di gradienti termici favorevoli al mantenimento del microclima. Le celle contenenti miele, polline, aria o pupe (covata) presentano differenti proprietà termiche (densità, calore specifico, conducibilità) che influenzano il flusso di calore. Modelli numerici hanno mostrato che una incorretta distribuzione dei materiali può generare gradienti termici irregolari che compromettono la stabilità del microclima [9]. Allo stesso tempo la combinazione e la corretta disposizione dei materiali può fungere da barriera termica naturale, isolando la zona della covata e rallentando le perdite verso l'ambiente esterno.

Anche il design e la struttura dell'alveare rientrano nelle misure passive per la termoregolazione, svolgendo un ruolo cruciale nella conservazione del calore. Si è osservato che i nidi naturali, generalmente localizzati in cavità d'albero alte, strette e con pareti spesse, offrono una resistenza termica superiore rispetto alle arnie artificiali in legno o polistirolo espanso (EPS), che risultano invece più soggette a dispersioni [2]. Le misurazioni sperimentali hanno dimostrato che la conduttanza termica delle arnie artificiali può essere da 4 a 7 volte superiore rispetto a quella dei nidi in albero, determinando una maggiore esposizione della colonia al freddo [2, 4].

Simulazioni CFD (*Computational Fluid Dynamics*) e modellazioni termiche mostrano inoltre che la forma verticale delle cavità arboree, combinata con una disposizione efficace del calore e dei materiali interni, riduce la necessità di "*clustering*", rendendo tale comportamento meno frequente, anziché essere pressoché obbligatorio come avviene in arnie artificiali poco isolate [3]. Questo significa che, in un ambiente termicamente efficiente come un nido naturale, le api possono distribuirsi in modo più funzionale, risparmiando energia che sarebbe altrimenti impiegata per il mantenimento del cluster.

I meccanismi di termoregolazione passiva costituiscono una vera e propria risorsa per contrastare l'elevata mortalità. A partire da queste osservazioni sarà sviluppato

nel presente progetto una serie di simulazioni CFD-CHT (*Conjugate Heat Transfer - Computational Fluid Dynamics*) con l'obiettivo di verificare, seppur in un caso semplificato, come alcuni dei meccanismi passivi influenzino la necessità o meno di innescare meccanismi termoregolatori attivi.

1.3 Tecnologie e soluzioni per il supporto alla termoregolazione

Per affrontare le difficoltà legate al mantenimento della temperatura invernale, sono stati sviluppati sistemi di monitoraggio ambientale basati su sensori elettronici, in grado di registrare in tempo reale parametri fondamentali per la sopravvivenza della colonia (temperatura, umidità e densità). Questi strumenti possono rilevare segnali precoci di collasso, come la riduzione dell'ampiezza termica interna, anche con 15 giorni di anticipo [4, 1]. Tuttavia, l'elevato costo e la necessità di manutenzione limitano la diffusione di queste tecnologie tra gli apicoltori. Soluzioni passive quali l'impiego di materiali a bassa conducibilità termica, (PET o sughero) hanno reso possibile il contenimento, delle perdite termiche, giungendo a risultati che si avvicinano a quelli registrati in cavità arboree [4]. Si rende tuttavia necessario in futuro un approfondimento per comprendere al meglio la dinamica dello scambio termico all'interno dell'alveare al fine di elaborare soluzioni passive ancora più efficaci, portando il tasso di mortalità ad essere il più basso possibile.

1.4 Modellazione numerica e simulazioni CFD

Negli ultimi anni, l'impiego di tecniche di modellazione numerica e *Computational Fluid Dynamics* (CFD) ha segnato un notevole avanzamento nello studio dei meccanismi termici all'interno degli alveari, permettendo di analizzare in dettaglio fenomeni complessi che sarebbero difficilmente misurabili con strumenti sperimentali.

Le simulazioni CFD sono state applicate con successo per modellare la distribuzione tridimensionale del calore all'interno del nido, evidenziando le dinamiche di accumulo e dispersione termica tra le celle, i materiali presenti (miele, polline, aria) e la covata. Questi modelli hanno permesso di descrivere come la disposizione dei materiali influenzi i gradienti termici e l'efficienza complessiva del mantenimento del microclima [10, 9].

Un altro importante ambito di ricerca riguarda la simulazione dei flussi d'aria indotti da *fanning* (ventilazione attiva) e ventilazione passiva, nonché i flussi convettivi generati dal metabolismo delle api. Le simulazioni CFD hanno permesso di quantificare il contributo di tali flussi alla regolazione di temperatura e umidità, e di identificare zone stagnanti o eccessivamente ventilate, che possono compromettere l'equilibrio termico del nido [10].

Questo studio prevede l'impiego di strumenti analoghi per l'analisi dello scambio termico, adottando quindi un approccio di tipo numerico. Nei paragrafi successivi verranno illustrati l'approccio seguito, i software utilizzati e la modellazione del caso di interesse.

Capitolo 2

Conjugate Heat Transfer (CHT): Principio e applicazioni ingegneristiche

Il *Conjugate Heat Transfer* (CHT) è un approccio di modellazione fisica utilizzato per analizzare il trasferimento di calore in sistemi fisici costituiti da solidi e fluidi. Questo fenomeno avviene in presenza di un gradiente termico, ossia una differenza di temperatura tra le varie parti del dominio. Tale fenomeno coinvolge la conduzione di calore nei materiali solidi, mentre convezione (naturale o forzata) e conduzione nei fluidi. L'utilizzo del modello CHT (*Conjugate Heat Transfer*) rispetto ad una classica simulazione CFD (*Computational Fluid Dynamics*) permette di simulare anche l'interazione termica tra i volumi fluidi e solidi. La regione del dominio fluido è governata dalle equazioni di Navier-Stokes, mentre il dominio solido dall'equazione della conduzione termica (Appendice A). Le applicazioni CHT in ambito ingegneristico sono innumerevoli (alcuni esempi riportati in [11], mentre può essere di interesse osservare il parallelismo riportato in [7], dove viene sfruttato questo approccio per confrontare il modello alveare con quello domestico). Obiettivo di questo lavoro è utilizzare lo stesso approccio CHT per modellare lo scambio termico in sistemi biologici complessi, gli alveari. In questi organismi il calore viene generato dalle api e trasferito attraverso la cera, il miele e l'aria interna, per il mantenimento di una condizione termica compatibile con la vita della colonia. A tal fine la modellazione CHT offre uno strumento potente per analizzare e simulare le dinamiche termiche in condizioni ambientali differenti, permettendo di valutare scenari ipotetici e progettare soluzioni ingegneristiche, oltre a consentire di approfondire la comprensione dei meccanismi di termoregolazione senza la necessità di un diretto approccio sperimentale. L'integrazione del CHT nella ricerca potrebbe quindi rappresentare una svolta nella progettazione di alveari sostenibili e termicamente efficienti. Ai fini dell'implementazione delle simulazioni CFD-CHT (*Computational Fluid Dynamics-Conjugate Heat Transfer*) sono stati utilizzati solver open-source: OpenFoam, CalculiX e la libreria preCICE, rispettivamente per la gestione del fluido, del solido e dell'interfaccia di scambio. Nei paragrafi successivi è riportata un'overview generale dei solver in questione.

2.1 OpenFOAM

OpenFOAM (Open Field Operation and Manipulation) è un solver open-source CFD che consente la simulazione di vari fenomeni associati al moto dei fluidi. La struttura di un generico case su Openfoam prevede l'implementazione di diverse directory, generalmente si ha:

- **system/** cartella contenenti diversi file che stabiliscono i parametri di controllo della simulazione, lo schema temporale, il setup per la creazione della mesh e le impostazioni adottate dal solutore per lo svolgimento dei calcoli e le verifiche sulla convergenza;
- **constant/** cartella contenente la mesh e i parametri fisici del fluido (viscosità, modelli di turbolenza e di trasporto, caratteristiche termofisiche, eventuali modelli di radiazione ecc.);
- **0/** cartella che definisce *IC* (condizioni iniziali) e *BC* (boundary conditions) per ciascun campo scalare o vettoriale (temperatura, velocità, pressione ecc.).

Nel presente lavoro OpenFoam è stato integrato ad altri software per modellare un caso di *Conjugate Heat Transfer*. Questa integrazione ha reso necessaria l'implementazione file aggiuntivi, deputati alla corretta interazione tra le interfacce di solido e fluido, descritti nel capitolo dedicato al *Coupling*.

2.2 CalculiX

CalculiX è un software open-source per l'analisi agli elementi finiti, FEA (*Finite Element Analysis*), sviluppato per simulazioni strutturali e termiche. Nel presente lavoro CalculiX è stato utilizzato per l'analisi e la risoluzione della componente solida della simulazione CHT (*Conjugate Heat Transfer*). Analogamente ad Openfoam, CalculiX prevede l'implementazione di directory e file deputati alla corretta computazione. In particolare si deve costruire la directory **case/solid/**, contenente il file (**.inp**) all'interno del quale vi sono tutte le informazioni necessarie per lo svolgimento dell'analisi computazionale (mesh, proprietà dei materiali, eventuali forze applicate, *BC*, step della simulazione ecc.). Inoltre si rende necessaria l'implementazione di un ulteriore file (**config.yml**) per la gestione del *Coupling*, anch'esso descritto nel capitolo dedicato e contenuto nella stessa directory **case/solid/**.

2.3 preCICE

preCICE (Precise Code Interaction Coupling Environment) è una libreria open-source progettata per la simulazione multifisica accoppiata avente il compito di facilitare l'interazione tra solver differenti. Il compito di questo tool è lo scambio di informazioni tra solido e fluido sfruttando: equation coupling, data mapping and communication [11].

La libreria preCICE è compatibile con diversi software open-source e commerciali, tra cui OpenFOAM e CalculiX, grazie a una serie di adapters dedicati (<https://precice.org/adapters-overview>). Nel presente lavoro PreCICE è stato

utilizzato per accoppiare OpenFOAM e CalculiX in una simulazione CHT multifisica, gestendo lo scambio di temperatura e flusso termico all'interfaccia tra fluido e solido. L'impiego della libreria preCICE ha richiesto l'implementazione di file di configurazione specifici (`precice-config.xml`) e l'integrazione di file lato solido e lato fluido per la comunicazione dei solver descritti nel capitolo dedicato al *Coupling*.

Capitolo 3

Modellazione

La fase iniziale del progetto di simulazione ha riguardato la modellazione: definizione dei domini con le semplificazioni del caso, l'annesso setup di configurazione, e l'elaborazione di modelli matematici.

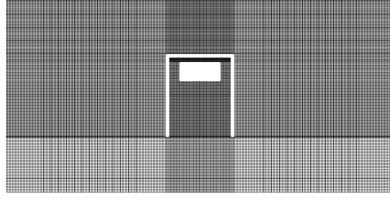
3.1 Fluid domain

La prima fase della modellazione ha riguardato la definizione del dominio fluido, adottando le semplificazioni opportune. Per l'analisi preliminare è stato considerato un dominio bidimensionale, ottenuto imponendo una sola cella lungo la direzione z . Al fine di semplificare ulteriormente il modello, la composizione del fluido è stata assunta uniforme sia all'interno che all'esterno dell'alveare, nonostante i processi metabolici interni ne modifichino inevitabilmente le caratteristiche. Il fluido è stato pertanto modellato come aria standard alla temperatura di 5°C . Sono stati inoltre trascurati gli effetti della radiazione solare (alveare idealmente posto in galleria del vento) e l'eventuale presenza di sorgenti di calore esterne aggiuntive. La sezione del dominio fluido è stata generata utilizzando lo strumento `blockMesh` di `OpenFOAM`, organizzata in modo da centrare il foro (regione priva di mesh) rispetto al dominio. In particolare, la regione destinata al solido è stata definita omettendo i corrispondenti blocchi nella sezione `blocks`, assicurando al contempo la corretta chiusura dei vertici.

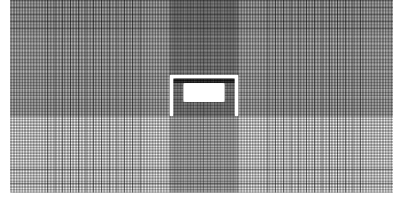
La dimensione delle celle è stata selezionata per ottenere un buon compromesso tra l'accuratezza fisica dei risultati e il Courant Number (CO), la cui importanza sarà discussa nella sezione dedicata ai risultati e al post-processing.

Le patch di interfaccia tra solido e fluido sono state definite separatamente nel `blockMeshDict` (pur trattando il solido come un'unica entità) al fine di agevolare le operazioni di post-processing.

Inoltre, in relazione a quanto descritto nella parte di letteratura si è voluto osservare come l'allungamento verticale dell'arnia influisca sulla quantità di calore disperso. Si è pertanto optato per modellare due domini fluidi nei quali si è considerato il caso standard a una cassetta ed il caso con due cassette sovrapposte Fig. 3.1.



(a) Cassetta doppia



(b) Cassetta singola

Figura 3.1: Mesh fluido nei due casi sopracitati. In entrambi i casi il dominio di fluido è lungo 2 m e alto 1 m ed è visibile il foro nel quale verrà allocato il solido

3.1.1 Modello di turbolenza

Il modello di turbolenza adottato per la simulazione in questione (specificato nel file `Fluid/constant/turbulenceProperties`) è il $k-\omega$ SST (Shear Stress Transport). La scelta è stata dettata dalla capacità del modello RANS in questione di gestione di flussi alle pareti in modo più efficace rispetto al $k-\epsilon$

In assenza di misure sperimentali all'ingresso, l'intensità di turbolenza (I) è stata stimata con la correlazione industriale per flussi interni $I = 0.16 \text{ Re}^{-1/8}$ [12]. Tale formula è pensata per flussi interni, il che rende la misurazione non pienamente applicabile ad un caso in ambiente esterno, tuttavia costituisce un'approssimazione necessaria in un primo luogo, nel quale si immagina la struttura solida in galleria del vento. Considerando aria a 5°C (temperatura esterna adottata per simulare una rigida condizione invernale) e 1 atm (valori di riferimento adottati nel caso: $\rho \approx 1.27 \text{ kg/m}^3$, $\mu \approx 1.75 \div 1.76 \times 10^{-5} \text{ Pa s}$, $Pr \approx 0.72$, $c_p \approx 1006 \text{ J/(kg K)}$), una lunghezza caratteristica $L = 0.356 \text{ m}$ e velocità media d'ingresso $U = 0.1 \text{ m/s}$, si ottiene $\text{Re} \approx 2.6 \times 10^3$ e dunque $I \approx 6\%$. Da ciò è possibile ricavare l'energia cinetica turbolenta $k = \frac{3}{2}(UI)^2 \approx 5.4 \times 10^{-5} \text{ m}^2/\text{s}^2$ e di conseguenza la frequenza specifica $\omega = \sqrt{k}/(C_\mu^{1/4} L_t) = 0.54 \text{ s}^{-1}$ (con $C_\mu = 0.09$ e $L_t = 0.07L$).

IC e BC campi turbolenti Nel dominio fluido, i campi da implementare nella directory `Fluid/0` per il modello di turbolenza adottato sono i seguenti, con la relativa scelta delle BC:

- *Energia cinetica turbolenta k* : Inizializzata con il valore calcolato analiticamente a partire dal numero di Reynolds. In termini di condizioni al contorno, si è imposto `fixedValue` all'inlet (garantendo un ingresso coerente), `zeroGradient` a outlet, top e bottom (permettendo il libero sviluppo senza riflessioni spurie), e `kqRWallFunction` sulle pareti di interfaccia `BroodHole` e `SolidHole`, in modo da assicurare il corretto decadimento di k nello strato limite.

- *Viscosità turbolenta cinematica* ν_t : Inizializzata con un valore uniforme molto piccolo (10^{-6} m²/s) come "seed" numerico. Le condizioni al contorno sono `calculated` per tutte le patch non di parete e `nutkWallFunction` sulle pareti dei solidi.
- *Diffusività turbolenta termica* α_t : Inizializzata a zero e trattata con `compressible::alphatWallFunction` su tutte le patch così da consentire di implementare un Prandtl turbolento tipico ($Pr_t \approx 0.85$). Ciò consente di modellare la corretta diffusione turbolenta del calore alle pareti.
- *Frequenza specifica* ω : Inizializzata con il valore trovato analiticamente, derivato da k, C_μ, L_t . Nel file di inizializzazione, ω è imposto come `fixedValue` all'inlet, `zeroGradient` a outlet, top e bottom, e con `omegaWallFunction` alle interfacce `BroodHole` e `SolidHole`, così da gestire ottimalmente l'analisi vicino alla parete.

3.1.2 buoyantPimpleFoam: file necessari e BC

Il solver `buoyantPimpleFoam` è stato scelto come motore fluidodinamico per la simulazione del dominio aria. Questo risolutore della libreria OpenFOAM integra le equazioni di Navier-Stokes in forma compressibile, includendo l'effetto di galleggiamento tramite il contributo idrostatico ρgh sfruttando $p_{rgh} = p^1 = p - \rho gh$, e risolve in parallelo l'equazione dell'energia (fondamentale per un accoppiamento CFD-CHT). La struttura dei file e le condizioni iniziali dei principali campi sono state definite con attenzione per garantire consistenza numerica e fisica con l'accoppiamento solido.

IC e BC campi necessari al solver Nella cartella 0/ sono stati implementati i file necessari (oltre a quelli per la turbolenza) per la risoluzione del dominio con `buoyantPimpleFoam` con le relative *BC* utilizzate:

- **T:** Il campo di temperatura è stato inizializzato con `internalField uniform 278.15`, (valore in accordo a quanto detto in precedenza, poichè OpenFoam opera in *Kelvin*). Alle pareti adiabatiche (top e bottom) è stato usato `zeroGradient`, mentre sulle patch di accoppiamento `BroodHole` e `SolidHole` è stato assegnato un `fixedGradient gradient 0` come "seed": queste condizioni, infatti, vengono poi sovrascritte dal coupling con il solido, lasciando a `preCICE` la gestione del flusso termico. All'inlet è stato imposto un `fixedValue` coerente, mentre all'outlet `zeroGradient`.
- **U:** Il campo di velocità è stato inizializzato a `internalField uniform (0 0 0)`. Al contorno inlet è stato definito un `fixedValue uniform (0.1 0 0)` per imporre una forzante al flusso, mentre all'outlet si è usato `zeroGradient` per consentire il deflusso. Alle pareti, invece, è stata applicata una condizione di `noSlip`.
- **p:** Il campo di pressione statica, come precedentemente accennato non è direttamente risolto dal solutore ma ricostruito come quantità derivata; per questo motivo si è utilizzato `type calculated` su tutte le patch, lasciando che sia OpenFOAM a calcolarla a partire da p_{rgh} .

- `p_rgh`: La pseudo pressione idrostatica è la variabile effettivamente risolta dal calcolatore. Qui sono state definite condizioni di `zeroGradient` a inlet e sulle pareti, mentre all'outlet è stato imposto un `fixedValue uniform $internalValue`.

3.2 Solid domain: modelli termo-fisici, condizioni al contorno e impostazioni numeriche

Il dominio solido è stato realizzato con l'utilizzo della combinazione di due programmi open-source.

La modellazione è stata inizialmente realizzata con FreeCAD per la creazione geometrica della struttura solida, successivamente esportata in PrePoMax, ambiente CalculiX friendly.

All'interno di PrePoMax è stata generata la mesh del dominio solido, definendo nodi, elementi e superfici, impostando inoltre condizioni iniziali sulla temperatura (294.15 K per il *Brood*, pari a 21 °C, e temperatura dell'ambiente esterno per il *Telaio*), BC necessarie, assegnazione dei materiali con rispettivi valori e definizione dello STEP della simulazione. Sono stati inoltre estratti i set di nodi e la superficie su cui avviene lo scambio termico (entrambi necessari per schema di accoppiamento adottato: nearest-neighbour, descritto in Appendice B). Il tipo di elemento utilizzato per la discretizzazione è stato C3D10: un elemento solido tetraedrico a 10 nodi (presenta i 4 nodi convenzionali degli elementi tetraedrici più uno su ogni segmento), adatto a simulazioni termo-strutturali con geometrie complesse (risulterebbe più efficace di altri specialmente se si volesse integrare al caso CHT anche FSI), poichè offre un'accuratezza superiore rispetto agli elementi tetraedrici a 4 nodi (C3D4) grazie ai nodi aggiuntivi che migliorano la rappresentazione delle deformazioni [13].

Infine, da PrePoMax è stato esportato il file `.inp`, per l'implementazione del coupling e dell'analisi computazionale. Il file comprensivo di tutte le informazioni è stato successivamente inserito nella directory `case/Solid/` secondo la struttura richiesta dal coupling con preCICE.

Per consentire un confronto diretto tra i risultati, come accennato in precedenza, sono stati definiti due domini distinti: il primo prevede una configurazione con cassetta singola, mentre il secondo una con cassetta doppia, al fine di analizzare come l'altezza del glomere dalla base dell'alveare influenzi la quantità di calore disperso. Sono quindi stati modellati due solidi differenti, le cui quote caratteristiche sono riportate in Fig. 3.2.

3.2.1 Proprietà termofisiche del solido.

Il dominio solido della simulazione è stato diviso in due regioni: il *Telaio*, struttura in legno (posta in equilibrio termodinamico con l'ambiente), modellata dalle pareti esterne, e la regione *Brood*, modellata come un materiale composito costituito da aria, api e cera (posto a 294.15 K temperatura ideale di mantenimento in condizione invernale in assenza di covata).

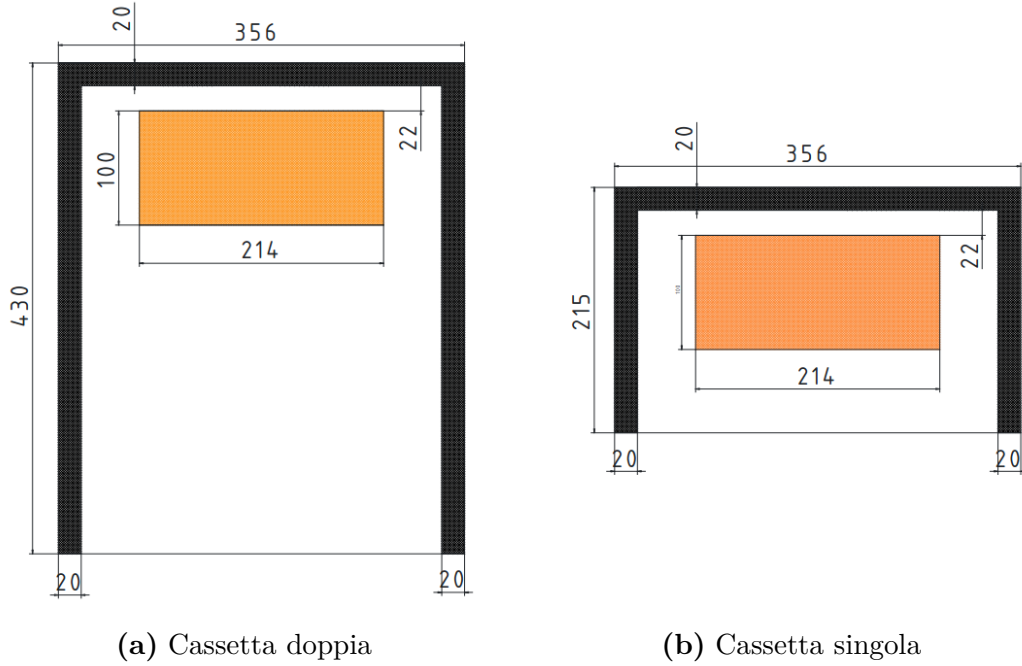


Figura 3.2: Solidi quotati: in nero è visibile la parte del *Telaio*, in arancione la parte del *Brood*: le misure per quest'ultimo sono state mantenute invariate per consentire una comparazione ragionevole

Legno Per la struttura esterna si è considerato un legno tipico per costruzione di arnie (abete). I valori di riferimento adottati sono:

$$\rho_{\text{legno}} = 420 \text{ kg/m}^3 \quad c_{p,\text{legno}} = 1530 \text{ J/(kg K)} \quad k_{\text{legno}} = 0.12 \text{ W/(m K)}$$

Queste proprietà ([14], per il c_p valore estremamente dipendente da tipologia di legno e umidità si è considerato un valore medio) sono considerate costanti nel tempo e nello spazio e sono state inserite in CalculiX tramite le carte `*DENSITY`, `*CONDUCTIVITY` e `*SPECIFIC HEAT`. Il telaio è stato modellato come una struttura unica, priva di porosità e di eventuali aperture per il ricircolo dell'aria, e rappresentato mediante una sezione a C simmetrica, ribaltata verso il basso. La parete superiore, costituita tipicamente dal coprifavo e dal tetto, è stata schematizzata semplificatamente come un unico blocco dello stesso spessore delle pareti laterali, pari a 20 mm.

Brood La regione del *Brood* è stata schematizzata semplificatamente come un rettangolo ed è trattata come materiale composito costituito da tre componenti (api, cera, aria, per le proprietà si è fatto riferimento a [9]):

- *Api*: $\rho_{\text{bee}} = 996 \text{ kg/m}^3$, $k_{\text{bee}} = 0.61 \text{ W/(m K)}$, $c_{p,\text{bee}} = 4180 \text{ J/(kg K)}$
- *Cera*: $\rho_{\text{wax}} = 950 \text{ kg/m}^3$, $k_{\text{wax}} = 0.40 \text{ W/(m K)}$, $c_{p,\text{wax}} = 2500 \text{ J/(kg K)}$
- *Aria* (5°C , 1 atm): $\rho_{\text{air}} = 1.1774 \text{ kg/m}^3$, $k_{\text{air}} = 0.02624 \text{ W/(m K)}$, $c_{p,\text{air}} = 1005.7 \text{ J/(kg K)}$

Le frazioni volumetriche sono state scelte pseudo arbitrariamente in quanto a priori è pressochè impossibile stabilire una composizione percentuale del glomere,

organismo biologico complesso in continuo cambiamento. Si è pertanto fatta un'ipotesi ragionevole stimando quanto segue, tuttavia si è costruito un modello replicabile con altre composizioni e/o frazioni volumetriche:

$$\phi_{\text{bee}} = 0.45, \quad \phi_{\text{wax}} = 0.30, \quad \phi_{\text{air}} = 0.25$$

Densità equivalente ρ_{eff} : Per un materiale composito con frazioni volumetriche ϕ_i e densità dei costituenti ρ_i , la massa totale contenuta in un volume è la somma delle masse parziali $\phi_i \rho_i$ (regola delle miscele), pertanto, per la densità:

$$\rho_{\text{eff}} = \sum_i \phi_i \rho_i$$

Da cui è possibile ricavare la densità equivalente del caso in esame

$$\rho_{\text{eff}} = \sum_i \phi_i \rho_i = 0.45 \cdot 996 + 0.30 \cdot 950 + 0.25 \cdot 1.1774 \approx 733.49 \text{ kg/m}^3$$

Calore specifico equivalente $c_{p,\text{eff}}$: Per il calore specifico a pressione costante, grandezza per unità di massa, si impone l'equivalenza dell'entalpia per unità di volume in condizioni di equilibrio termico locale, ovvero: $\rho_{\text{eff}} c_{p,\text{eff}} T = \sum_i \phi_i \rho_i c_{p,i} T$, si ottiene pertanto una media sulle frazioni massiche $m_i = \phi_i \rho_i / \rho_{\text{eff}}$:

$$c_{p,\text{eff}} = \frac{\sum_i \phi_i \rho_i c_{p,i}}{\sum_i \phi_i \rho_i} = \sum_i m_i c_{p,i}, \quad m_i = \frac{\phi_i \rho_i}{\rho_{\text{eff}}}$$

E' pertanto possibile ricavare

$$m_i = \frac{\phi_i \rho_i}{\rho_{\text{eff}}} \Rightarrow m_{\text{bee}} \approx 0.61105, \quad m_{\text{wax}} \approx 0.38855, \quad m_{\text{air}} \approx 4.013 \times 10^{-4}$$

Da cui è possibile ricavare il calore specifico efficace come la media pesata sulle m_i :

$$c_{p,\text{eff}} = \sum_i m_i c_{p,i} \approx 0.61105 \cdot 4180 + 0.38855 \cdot 2500 + (4.013 \times 10^{-4}) \cdot 1005.7 \approx 3526.0 \text{ J/(kg K)}$$

Conducibilità termica equivalente k_{eff} : L'ultimo parametro termicamente rilevante per il solido è costituito dalla conducibilità termica. Per lo studio di questa grandezza in un materiale composito si è adottato il modello di Bruggeman [15]:

$$\sum_{i=1}^3 \phi_i \frac{k_i - k_{\text{eff}}}{k_i + 2k_{\text{eff}}} = 0, \quad \sum_{i=1}^3 \phi_i = 1$$

dove ϕ_i sono, come già visto, le frazioni volumetriche e k_i le conducibilità associate ai rispettivi materiali. Ponendo per semplicità $x = k_{\text{eff}}$ ed eliminando il denominatore (moltiplicando ambo i membri per $\prod_{m=1}^3 (k_m + 2x)$) si ottiene:

$$\sum_{i=1}^3 \phi_i (k_i - x) \prod_{j \neq i} (k_j + 2x) = 0$$

che è un'equazione cubica in x . Si sono definite in seguito le somme simmetriche $S_1 = k_1 + k_2 + k_3$, $S_2 = k_1 k_2 + k_2 k_3 + k_3 k_1$, $S_3 = k_1 k_2 k_3$ e i momenti pesati $A_1 = \sum_i \phi_i k_i$, $A_2 = \sum_i \phi_i k_i^2$. Dopo lo sviluppo si ottiene la forma compatta:

$$-4x^3 + \underbrace{(6A_1 - 2S_1)}_{=C_2} x^2 + \underbrace{(-S_2 + 3S_1 A_1 - 3A_2)}_{=C_1} x + \underbrace{S_3}_{=C_0} = 0$$

moltiplicando per -1 :

$$4x^3 - C_2 x^2 - C_1 x - C_0 = 0$$

considerando i dati del problema Con $\phi_{\text{bee}} = 0.45$, $\phi_{\text{wax}} = 0.30$, $\phi_{\text{air}} = 0.25$ e $k_{\text{bee}} = 0.61 \text{ W/(m K)}$, $k_{\text{wax}} = 0.40 \text{ W/(m K)}$, $k_{\text{air}} = 0.02624 \text{ W/(m K)}$, è possibile calcolare:

$$S_1 = 1.03624, \quad S_2 = 0.2705024, \quad S_3 = 0.00640256, \quad A_1 = 0.40106, \quad A_2 = 0.215617$$

Da cui si ricavano i coefficienti:

$$C_2 = 6A_1 - 2S_1 = 0.33388, \quad C_1 = -S_2 + 3S_1 A_1 - 3A_2 = -0.32942944, \\ C_0 = S_3 = 0.00640256$$

e la cubica risulta pertanto essere:

$$4x^3 - 0.33388 x^2 - 0.32942944 x - 0.00640256 = 0$$

Per trovare una soluzione in forma chiusa si è sfruttato il metodo di Cardano per la risoluzione delle cubiche; dividendo per 4 si ottiene la forma monica dell'equazione

$$x^3 + px^2 + qx + r = 0$$

ove:

$$p = -0.08347, \quad q = -0.08235736, \quad r = -0.00160064$$

Si elimina il termine quadratico con $x = y - \frac{p}{3}$, ottenendo la cubica $y^3 + Py + Q = 0$ con

$$P = q - \frac{p^2}{3} \approx -0.0846797736, \quad Q = \frac{2p^3}{27} - \frac{pq}{3} + r \approx -0.00393517817$$

Poiché $\Delta = \left(\frac{Q}{2}\right)^2 + \left(\frac{P}{3}\right)^3 < 0$, si applica la forma trigonometrica (casus irreducibilis):

$$y = 2\sqrt{-\frac{P}{3}} \cos\left(\frac{1}{3} \arccos \frac{-Q/2}{\sqrt{(-P/3)^3}}\right), \quad x = y - \frac{p}{3}$$

Sostituendo i valori di P e Q si ottiene l'unico radice fisicamente ammissibile in quanto l'unica positiva

$$k_{\text{eff}} = x \approx 0.3397458334 \text{ W/(m K)} \approx 0.340 \text{ W/(m K)}$$

Verifica di consistenza del modello su diffusività termica α . Per verificare la consistenza del modello si è calcolata la diffusività termica:

$$\alpha = \frac{k_{\text{eff}}}{\rho_{\text{eff}} c_{p,\text{eff}}} = \frac{0.33975}{733.49 \times 3526.0} \approx 1.31 \times 10^{-7} \text{ m}^2/\text{s},$$

tale valore è in accordo a quello di materiali con discrete proprietà isolanti, in analogia a come si auspicava [16].

Capitolo 4

Coupling

L'accoppiamento tra il dominio fluido (gestito da OpenFOAM) e quello solido (gestito da CalculiX) è stato realizzato tramite la libreria preCICE, progettata specificamente per facilitare la comunicazione tra solver differenti.

Nel caso in esame, sono stati utilizzati i preCICE adapter per OpenFOAM e CalculiX, che permettono l'integrazione della libreria all'interno dei rispettivi solver. Durante la simulazione accoppiata, preCICE gestisce i due domini (temperatura e flusso termico nel caso in esame), assicurando un continuo scambio di dati e garantendo il sincronismo temporale tra i due solver (il timestep dei solver fluido e solido si "adattano" automaticamente a quello indicato nel `.xml`).

Affinché l'accoppiamento avvenga correttamente, è necessaria l'implementazione di una serie di file aggiuntivi rispetto a quelli tipici di una simulazione tradizionale CFD, vale a dire:

4.1 preciceDict

Lato fluido, l'accoppiamento con preCICE richiede la creazione del file `preciceDict` collocato in `case\Fluid\system\`. Questo file dichiara le patch accoppiate su cui avviene lo scambio termico:

Nel file vengono specificati:

- il percorso al file di configurazione principale di preCICE, `precice-config.xml`, contenente i parametri del coupling;
- il nome del partecipante nel coupling (`participant`), che deve coincidere con quello dichiarato nel `precice-config.xml`
- il modulo di accoppiamento utilizzato (in questo caso CHT, *Conjugate Heat Transfer*);
- la mesh di interfaccia definita dal lato fluido, identificata da un nome: `Fluid-Mesh` (che deve coincidere con quello dichiarato nel `precice-config.xml`);
- la lista delle patch associate all'interfaccia di cui sopra (nel caso in esame, le patch `SolidHole` `BroodHole`);
- i dati in ingresso per OpenFOAM (`readData Heat-Flux`) in accordo al `precice-config.xml`;

- i dati in uscita da OpenFOAM (`writeData Temperature`) in accordo al `precice-config.xml`.

4.2 config.yml

L'accoppiamento del solido con preCICE richiede, invece, la creazione del file `config.yml` collocato in `case\Solid\`. Questo file mette in comunicazione il solver (CalculiX) e la libreria preCICE, abilitando lo scambio termico. Il file deve contenere:

- il nome del partecipante nel coupling (`participant`), che deve coincidere con quello dichiarato nel file `.xml`
- le interfacce (poichè si è utilizzata la condizione al contorno di Dirichlet-Neuman è necessario definire due set, uno di nodi e l'altro delle supefici su cui avviene lo scambio [11]);
- la mesh dei nodi (`Solid-Mesh-nodes`) sulla quale il solutore riceve i dati (`read-data Temperature`), il cui nome dovrà essere in accordo con quanto dichiarato nel `precice-config.xml`.
- la patch associata all'interfaccia di cui sopra, così come è definita nel file `.inp`;
- la grandezza ricevuta dal partecipante fluido (`Temperature`), ricevuta in corrispondenza dei nodi.

Analogamente si procede con la superficie su cui avviene lo scambio `flux_interface` ove verranno scritti i dati dell'`Heat-Flux` successivamente inviati al fluido.

4.3 precice-config.xml

L'ultimo file essenziale per la corretta esecuzione della simulazione accoppiata è il file `precice-config.xml`, all'interno del quale vengono definite in modo esplicito tutte le informazioni necessarie al coupling tra i partecipanti. Questo file deve essere visibile ad entrambe le directory `case/Fluid` e `case/Solid`, pertanto sarà collocato in `case` e costituisce l'aspetto principale del sistema di accoppiamento di preCICE, poiché specifica:

- le quantità fisiche scambiate tra i domini (devono essere in accordo a quanto dichiarato nel `preciceDict` e nel `config.yml`);
- le mesh di interfaccia (devono essere in accordo a quanto dichiarato nel `preciceDict` e nel `config.yml`);
- i metodi di interpolazione utilizzati;
- il timestep della simulazione;
- una serie di parametri trattati maggiormente in Appendice B (schema di accoppiamento, accelerazione, parametri di convergenza ecc)

N.B. è fondamentale che le mesh dichiarate nei rispettivi file implementati per solido e fluido coincidano con quelle dichiarate nel `precice-config.xml`. Per una migliore comprensione dei file si consiglia di visionare la sezione *Tutorial* in Appendice B e il git-hub associato al caso.

Capitolo 5

Risultati e post-processing

5.1 Visualizzazione dei campi di temperatura e velocità

Le simulazioni sono state svolte su un tempo complessivo di 30 min. Per l'analisi dei campi di temperatura e velocità è stato considerato l'ultimo *timestep* significativo della simulazione, corrispondente a 1800 s (ossia 30 min). Le immagini, ottenute mediante post-processing con ParaView, evidenziano come in entrambi i casi, trascorso tale intervallo temporale, il *Brood* abbia perso parte del calore inizialmente presente. I meccanismi di conduzione e convezione hanno quindi propagato l'energia termica, contribuendo al riscaldamento delle pareti interne del telaio Fig 5.1.

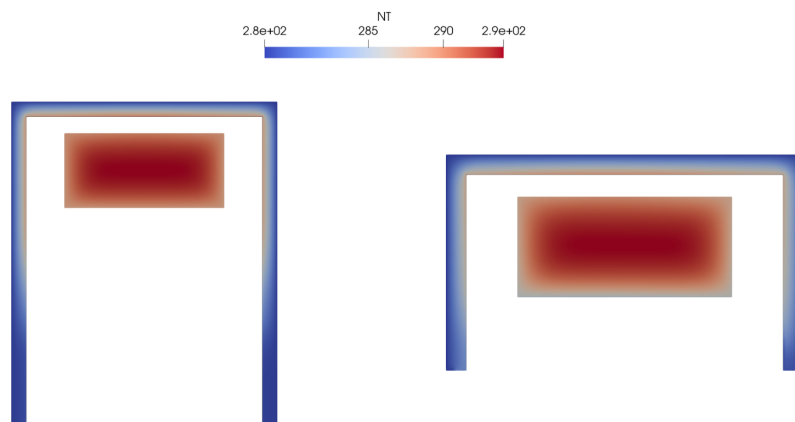


Figura 5.1: Visualizzazione del telaio solido riscaldato trascorsi 1800 s: con NT è indicata la temperatura dei nodi del solido.

Per quanto riguarda il campo di velocità, si osserva la presenza di un punto di ristagno in corrispondenza della mezzeria del solido, dove il flusso impatta direttamente sulla superficie. Il flusso si adatta quindi alla geometria, generando una bolla di ricircolo in prossimità del tetto del telaio. A valle, dopo un transitorio

iniziale, si formano due vortici controrotanti, più marcati nella configurazione a doppia cassetta e variabili in intensità e direzione in quella a cassetta singola Fig. 5.2. Nella configurazione a cassetta doppia questi ultimi risultano leggermente inclinati verso il basso a causa dell'asimmetria del solido rispetto all'asse x e del pronunciato sviluppo verticale se paragonato alla dimensione del dominio.

Sono state inoltre tracciate, in colore bianco, le isoterme corrispondenti a differenti temperature. Queste mostrano un andamento pressoché simmetrico rispetto all'asse y , ad eccezione delle isoterme più basse, deviate dall'azione del campo di velocità. Si osserva inoltre che, nella configurazione a cassetta singola, parte del flusso riscaldato viene trasportato verso l'esterno dal moto convettivo del flusso, mentre nella configurazione a doppia cassetta il calore rimane confinato e viene trasferito in maniera più completa al telaio.

Infine, si rileva un comportamento differente della scia: nella configurazione a cassetta singola, la ridotta dimensione in direzione y induce un'oscillazione tipica dei corpi tozzi investiti da flusso, mentre nella configurazione a cassetta doppia, a causa delle dimensioni più limitate del dominio, il flusso separa ma non riesce a richiudersi a valle del solido. Ciò determina la formazione di due grandi vortici controrotanti stazionari in prossimità della parete esterna destra del telaio.

5.2 Time window e stabilità numerica

Time window e convergenza. La `time-window-size` è stata fissata a $\Delta t = 0.1$ s, pari al passo temporale del solido, garantendo la sincronizzazione perfetta fra i due domini accoppiati. La scelta di Δt è stata guidata da considerazioni di stabilità numerica: nelle simulazioni accoppiate di tipo CHT (Conjugate Heat Transfer) è necessario mantenere il numero di Courant al di sotto di una soglia critica:

$$CO = \frac{U \Delta t}{\Delta h} < 15,$$

(U : velocità del flusso, Δt : time step caratteristico della simulazione, Δh : dimensione caratteristica monodimensionale di una cella computazionale della mesh) al fine di assicurare stabilità numerica durante la simulazione. Il CO è un parametro fondamentale da osservare nelle simulazioni CFD, rappresenta infatti la frazione della cella che il fluido attraversa in un singolo time-step. Per questa particolare simulazione si è osservato che un $CO < 15$ assicura un buon condizionamento, tenendo anche conto delle limitazioni imposte dalle risorse computazionali. Questo parametro potrebbe pertanto richiedere un adattamento in base alle condizioni fisiche impostate (es variando conducibilità e calori specifici), rendendo talvolta necessario un riadattamento della sezione `blocks` del `blockMeshDict` o del timestep.

Funzionalità dello script Python: `CO.py` Lo script esegue le seguenti operazioni:

1. Estrae dal `solver.log` i valori *Courant Number mean/max* ad ogni timestep.
2. Associa ad ogni t i relativi valori ($t, CO_{\text{mean}}, CO_{\text{max}}$).
3. Traccia un grafico con CO_{mean} e CO_{max} in funzione del tempo, utile per verificare la stabilità numerica della simulazione.

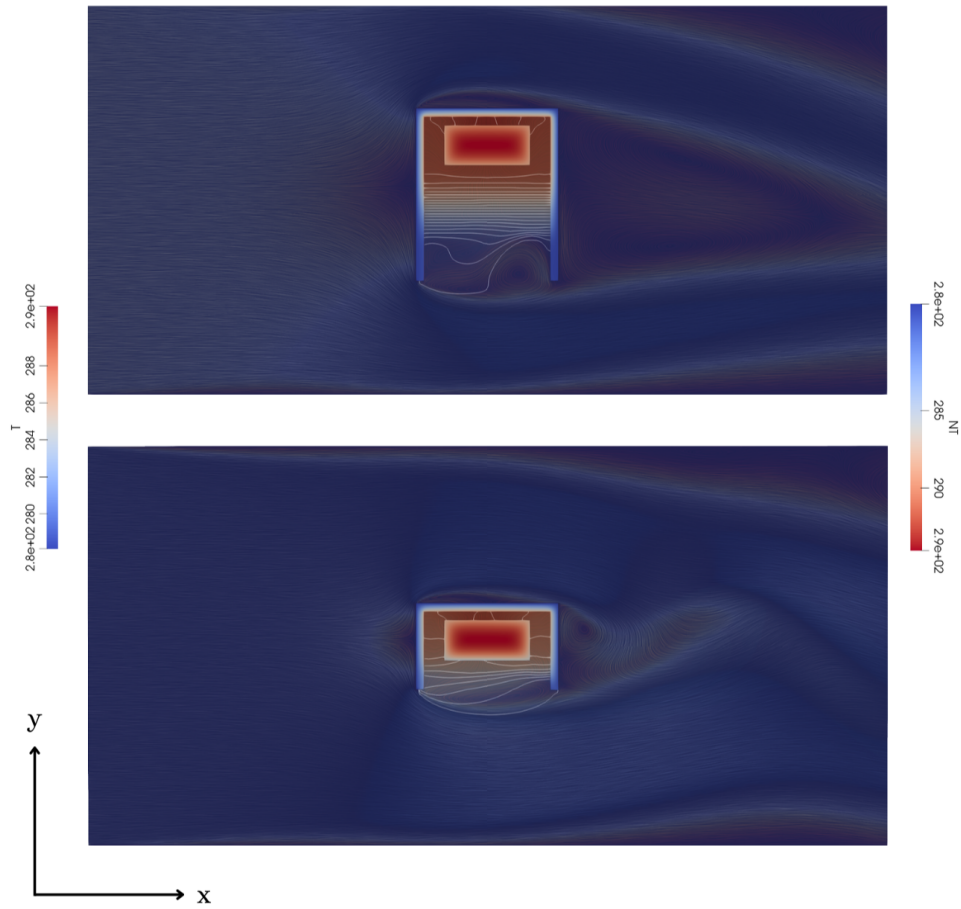


Figura 5.2: Campo di temperatura e linee di flusso nel caso a cassetta doppia (sopra) e cassetta singola (sotto): con T è indicata la temperatura del fluido, mentre con NT la temperatura dei nodi del solido, in bianco sono rappresentate le isoterme del fluido

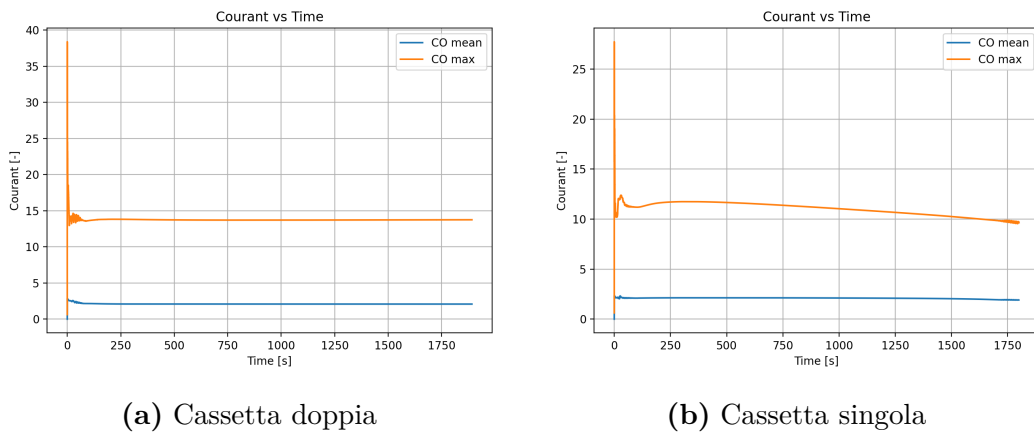


Figura 5.3: Grafico del *Courant Number* vs *Time* nel caso a cassetta doppia (a) e singola (b)

Considerazioni In entrambi i casi analizzati (Fig. 5.3) si è osservato che, dopo un transitorio iniziale di pochi secondi in cui il problema risulta mal condizionato, il valore del $max(CO)$ si stabilizza su livelli accettabili per entrambe le simulazioni. Nella configurazione con due alveari sovrapposti tale valore risulta leggermente più elevato, poiché il numero di nodi coinvolti nello scambio termico (e di conseguenza le velocità dovute al galleggiamento) aumenta a causa delle dimensioni del solido maggiori.

5.3 Funzioni nel `controlDict`.

La configurazione del file `system/controlDict`, con l'implementazione di alcune function è stata fondamentale per l'analisi dei dati, in particolare si è voluto analizzare l'andamento del calore trasferito da `BroodHole` nel fluido. A tal proposito sono state implementate delle function, in particolare:

- **wallHeatFlux**: questa funzione calcola il *flusso termico locale di parete* $q_w(\mathbf{x})$ [W/m^2] sulle patch specificate. La definizione segue la legge di Fourier:

$$q_w(\mathbf{x}) = -k_{\text{eff}} \left. \frac{\partial T}{\partial n} \right|_{\Gamma},$$

dove k_{eff} è la conducibilità termica effettiva, T è la temperatura e $\frac{\partial}{\partial n}$ è la derivata normale uscente alla superficie Γ . Si tratta quindi di un campo distribuito, espresso in termini di densità di flusso (potenza per unità di area).

- **qIn_BroodHole** e **qIn_SolidHole**: tali funzioni, configurate come `surfaceFieldValue` con operazione `areaIntegrate`, eseguono l'integrale superficiale del flusso locale calcolato da `wallHeatFlux`. In formula:

$$\dot{Q}_{\Gamma} = \int_{\Gamma} q_w(\mathbf{x}) dA,$$

dove Γ corrisponde al patch `BroodHole` o `SolidHole`. L'uscita è quindi una potenza termica totale [W] scambiata attraverso ciascuna superficie di contatto. In questo modo si dispone sia del campo locale $q_w(\mathbf{x})$, utile per analisi distribuite, sia dei valori globali $\dot{Q}_{\Gamma}$, necessari a bilanci energetici e confronti quantitativi.

5.3.1 Analisi del flusso termico uscente da `BroodHole` nel tempo

In primo luogo si è voluto graficare l'andamento del calore scambiato dal solido a una temperatura maggiore (*Brood*) rispetto al dominio con il fluido. A tal proposito si è considerata la potenza istantanea scambiata e il calore in funzione del tempo (come integrale su t della potenza)

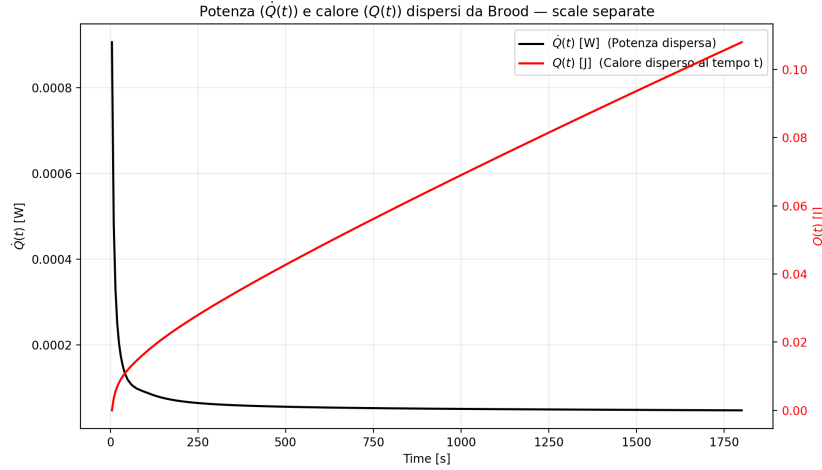
Funzionalità dello script Python: `HF.py` Lo script esegue le seguenti operazioni:

1. estrae i dati del flusso termico istantaneo dai file `surfaceFieldValue.dat` associati a `BroodHole`

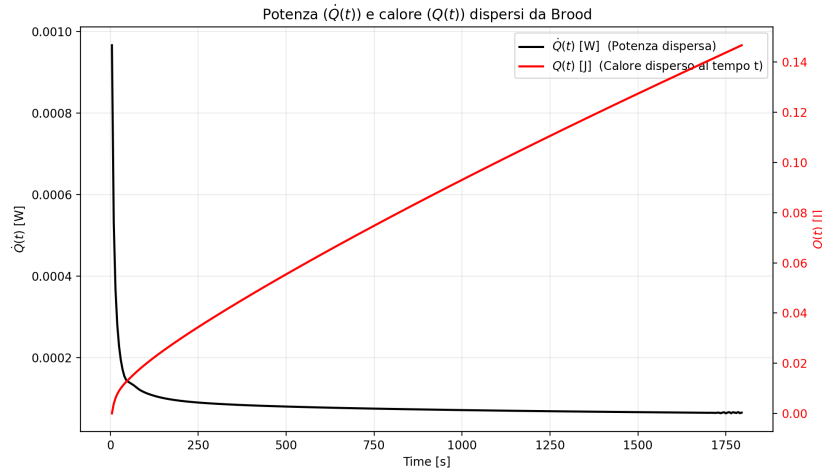
2. calcola l'integrale numerico sfruttando il metodo dei trapezi ($\Gamma = \text{Brood}$):

$$Q(t) = \int_0^t \dot{Q}_\Gamma(\tau) d\tau.$$

3. grafica la potenza istantanea $\dot{Q}_\Gamma(t)$ [W] e l'energia dispersa dal solido al tempo t , $Q(t)$ [J].



(a) Cassetta doppia



(b) Cassetta singola

Figura 5.4: Grafico del calore e della potenza vs *time* dispersi da Brood nel caso a cassetta doppia (a) e singola (b).

Considerazioni Dall'analisi dei grafici (Fig. 5.4), confrontando la quantità di calore dispersa a parità di tempo, la configurazione a cassetta doppia risulta preferibile. Si osserva, infatti, che a parità di t presenta una dissipazione inferiore e, quindi,

una migliore conservazione delle condizioni termiche ottimali per la sopravvivenza della colonia.

Si osserva, infatti, che al tempo $t = 30$ min si osserva che

$$Q_{\text{singola}} \approx 0.15 \text{ J}, \quad Q_{\text{doppia}} \approx 0.11 \text{ J}.$$

La riduzione percentuale della dissipazione nella cassetta doppia (calcolata rispetto al caso singolo) è

$$\xi (\%) = \frac{Q_{\text{singola}} - Q_{\text{doppia}}}{Q_{\text{singola}}} \times 100 = \frac{0.15 - 0.11}{0.15} \times 100 = \frac{0.04}{0.15} \times 100 \approx 26.67\%.$$

Pertanto, a 30 min la cassetta doppia dissipa circa il 26.7% in meno rispetto alla cassetta singola.

Si è provato, pertanto, che anche nel caso in esame in accordo a quanto riportato in letteratura, uno sviluppo verticale dell'alveare favorisce un periodo più lungo di mantenimento delle condizioni di temperatura ideale, risultando più efficace della configurazione a cassetta singola.

5.3.2 Comparazione andamento del flusso sperimentale e teorico

Dalla visualizzazione dei grafici si è notato un andamento vicino a quello auspicabile secondo i modelli tradizionali teorici per il fenomeno dell'Heat transfer, pertanto si è voluto comparare quanto ottenuto numericamente (in termini di potenze disperse) con i modelli teorici proposti per lo scambio termico. Per visualizzare ciò si sono adottate le seguenti rappresentazioni:

- *Scala semilogaritmica*, utile a verificare un eventuale decadimento esponenziale:

$$\dot{Q}(t) = Q_0 e^{-t/\tau},$$

che in un grafico $\log(\dot{Q})$ vs t si manifesta come una retta, con τ tempo caratteristico dello scambio.

- *Dipendenza diffusiva in semispazio semi-infinito*, dove il flusso segue la legge:

$$\dot{q}''(t) = \frac{k \Delta T}{\sqrt{\pi \alpha t}}, \quad \alpha = \frac{k}{\rho c_p},$$

con k conducibilità, α diffusività termica e ΔT salto di temperatura iniziale. Integrando su area A :

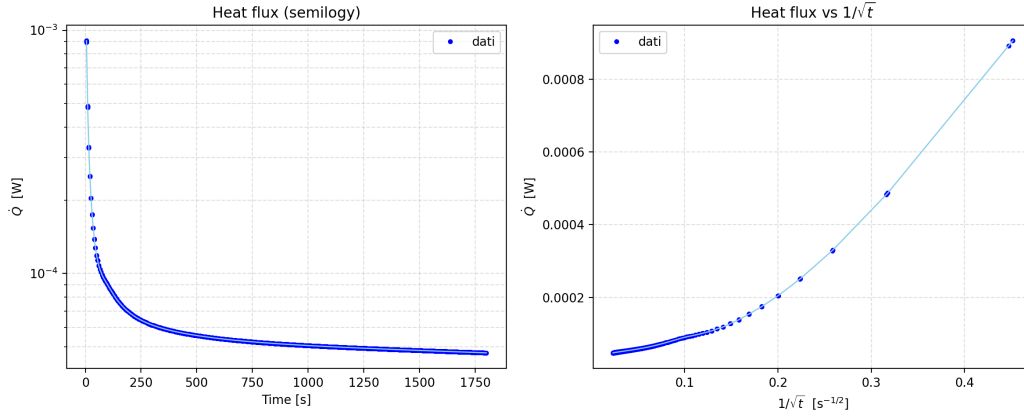
$$\dot{Q}(t) = A \dot{q}''(t) \propto \frac{1}{\sqrt{t}}.$$

In un grafico $\dot{Q}(t)$ vs $1/\sqrt{t}$ ci si attende un andamento lineare nel regime iniziale, con eventuali deviazioni dovute a effetti di spessore finito, convezione o variazioni di ΔT .

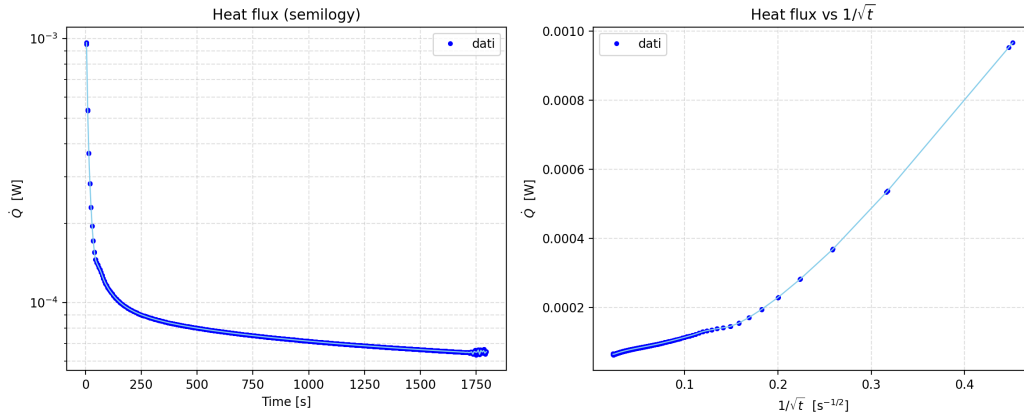
Il confronto tra le due viste consente di discriminare se lo scambio termico è dominato da un decadimento esponenziale (singolo tempo caratteristico τ) oppure dal regime diffusivo classico (conduzione transiente).

Funzionalità dello script Python: `fit_HF_views.py` Lo script esegue le seguenti operazioni:

1. estrae i dati associati alla potenza termica istantanea dispersa $\dot{Q}(t)$ estraendoli da `surfaceFieldValue` e li associa ai relativi timestep;
2. li rappresenta in scala semilogaritmica ($\log(\dot{Q})$ vs t) e come funzione di $1/\sqrt{t}$ interpolando i dati con una curva.



(a) Cassetta doppia



(b) Cassetta singola

Figura 5.5: Analisi delle potenze in scala semilogaritmica e $\propto 1/\sqrt{t}$. In blu sono rappresentati i dati numerici, mentre in azzurro la curva interpolante

Considerazioni In entrambi i casi analizzati (Fig. 5.5) si è osservato che l'andamento non coincide esattamente con nessuno dei due modelli teorici. Ciò è probabilmente imputabile all'elevato livello di semplificazione introdotto nel problema e ai limiti intrinseci dei modelli teorici rispetto al caso reale; nel primo caso si ha un modello esponenziale che spesso presenta disallineamenti dal modello reale, mentre nel secondo caso il modello di Fourier che per domini di dimensione finita

presenta limitazioni notevoli. Dal grafico di `HF.py` emerge tuttavia come l'andamento sperimentato si collochi verosimilmente tra i due modelli, il che consente di ritenere plausibili e qualitativamente affidabili i dati ottenuti.

5.3.3 Confronto tra potenza generata dalle api e potenza termica uscente

Si è successivamente confrontata la potenza termica generata dalle api con quella dispersa attraverso la patch `BroodHole`. Si è preferito utilizzare la patch `BroodHole` anziché `SolidHole` in quanto nel caso a cassetta singola parte del calore viene disperso per il moto convettivo del flusso, anziché trasmesso a `SolidHole` e poi disperso. A questo fine è stato sviluppato un modello che simuli la potenza generata dalle api ($\dot{Q}_{\text{bees}}$) per compararlo al flusso termico uscente ($\dot{Q}_{\text{out}}(t)$) dalla patch.

Modello costante di potenza delle api. Per la creazione di questo modello si è considerato il caso semplificato nel quale la produzione di calore sia costante, sia indipendente dal consumo metabolico e coinvolga tutte le api presenti (considerando il valore massimo teorico di produzione di calore e ignorando le dinamiche di compresenza di più api riscaldatrici). Pertanto, avendo considerato in precedenza $\phi_{\text{bee}} = 0.45$ la frazione volumetrica di api nel solido, ρ_{bees} la loro densità, q_{bee} la potenza specifica per unità di massa (W/kg) e L_x, L_y, L_z le dimensioni del dominio solido. Il volume complessivo è

$$V = L_x L_y L_z$$

il volume occupato dalle api

$$V_{\text{bee}} = \phi_{\text{bee}} V$$

e quindi la massa

$$m_{\text{bees}} = \rho_{\text{bee}} V_{\text{bee}} = \rho_{\text{bee}} \phi_{\text{bee}} V$$

Assumendo la potenza specifica costante, la potenza totale generata è

$$\dot{Q}_{\text{bees}} = q_{\text{bee}} m_{\text{bees}} = q_{\text{bee}} \rho_{\text{bee}} \phi_{\text{bee}} L_x L_y L_z$$

Con i valori utilizzati nello script:

$$L_x = 0.214 \text{ m}, \quad L_y = 0.100 \text{ m}, \quad L_z = 0.001 \text{ m},$$

$$\phi_{\text{bee}} = 0.45, \quad \rho_{\text{bee}} = 996 \text{ kg/m}^3, \quad q_{\text{bee}} = 500 \text{ W/kg}$$

risulta

$$\dot{Q}_{\text{bees}} \approx 4.80 \text{ W}$$

Funzionalità dello script Python: `bees_vs_BroodHole.py` Lo script esegue le seguenti operazioni:

1. Calcola analiticamente $\dot{Q}_{\text{bees}}$ secondo il modello di cui sopra.
2. Legge i file `surfaceFieldValue.dat` associati a `BroodHole`, estraendo le coppie $(t, \dot{Q}_{\text{out}}(t))$.

3. Ordina i dati per tempo e costruisce il vettore temporale e la potenza uscente.
4. Confronta $\dot{Q}_{\text{out}}(t)$ con $\dot{Q}_{\text{bees}}$, calcolando gli eventuali tempi di intersezione, ossia i valori t^* tali che

$$\dot{Q}_{\text{out}}(t^*) = \dot{Q}_{\text{bees}}.$$

5. Produce due grafici che riportano:

- Il confronto tra le energie: (J) $Q_{\text{bees}} = \dot{Q}_{\text{bees}}(t) t$ e $Q_{\text{Brood}}(t) = \int_0^t \dot{Q}_{\text{Brood}}(\tau) d\tau$.
(calcolo di Q_{Brood} analogo a quello ricavato con `HF.py`);
- Il confronto tra potenza delle api $\dot{Q}_{\text{bees}} \approx 4.80 \text{ W}$ e energia del *Brood*, Q_{Brood} ;

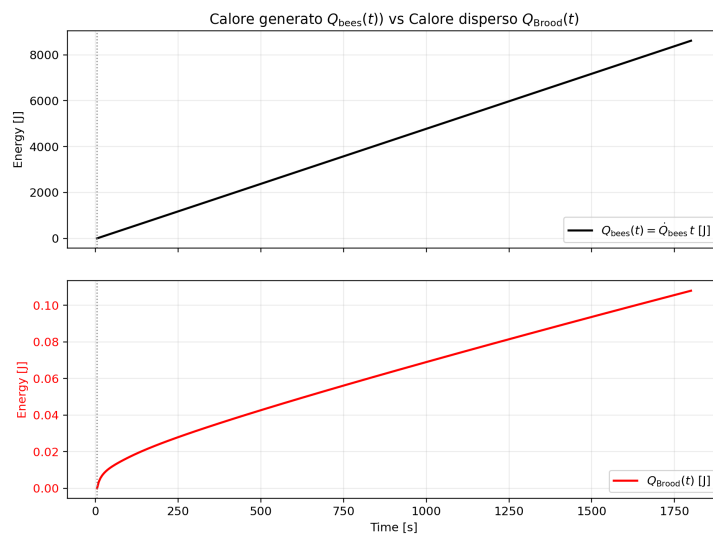
Considerazioni Per una corretta interpretazione dei grafici è opportuno sottolineare che il tempo di simulazione è stato limitato a 30 min, a causa dei vincoli di potenza di calcolo e di memoria disponibile. Inoltre, occorre evidenziare che in un grafico (Fig. 5.7) vengono confrontate due grandezze di natura differente: la potenza erogata dalle api (espressa in Watt) e il calore disperso al tempo t (espresso in Joule).

Questo è motivato dal fatto che se si osserva solamente l'andamento del calore non si riesce ad apprezzare un risultato fisicamente realistico, infatti, il calore disperso dal solido segue un andamento compreso tra una crescita esponenziale e un andamento di tipo $1/\sqrt{t}$, mentre il calore generato dalle api, ipotizzando un modello a potenza costante, cresce linearmente nel tempo. Ne consegue che il contributo riscaldante delle api risulta sempre maggiore rispetto al calore disperso come è apprezzabile dal grafico Fig. 5.6. Il confronto pertanto dei soli calori conduce a risultati non realistici.

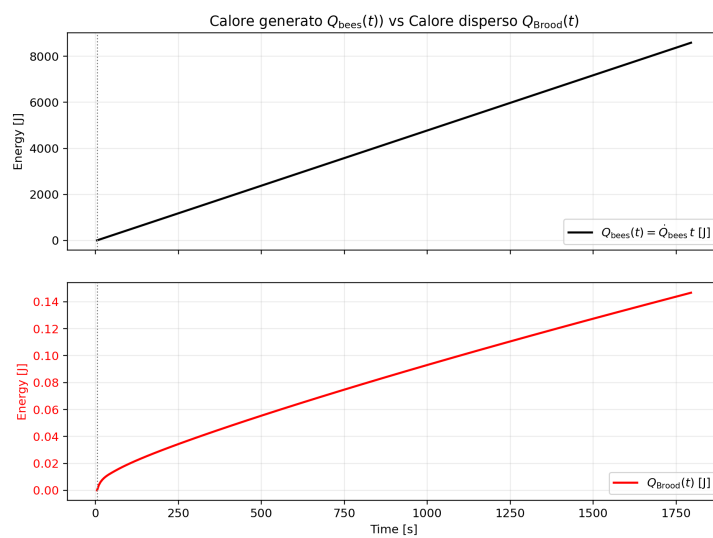
Tale risultato, deriva in grande parte dal modello irrealistico della generazione di potenza costante (tutte le api contribuiscono attivamente al riscaldamento, non viene considerato il consumo metabolico, la potenza erogata dalle api è assunta costante nel tempo ecc.).

Queste approssimazioni comportano inevitabili discrepanze rispetto alla realtà, la quale si avvicina di più a uno scenario in cui le api non iniziano immediatamente a riscaldare la covata con la massima efficienza. È quindi plausibile ritenere che la colonia (in particolare in alveari bassi) inizi la termoregolazione poco prima che le curve di potenza e calore si intersechino.

Se si desidera stimare l'ordine di grandezza dell'intersezione delle curve in questo secondo caso possiamo considerare in prima approssimazione un andamento lineare della dispersione termica: nel caso a singola cassetta una perdita di circa 0.15 W in 30 min corrisponde 4.8 W in 16 ore, nel caso a cassetta doppia una perdita di circa 0.11 W in 30 min corrisponde 4.8 W in 21.8 ore . Questi valori hanno un carattere prettamente qualitativo e non devono essere usati come metro di paragone, tuttavia consentono di inquadrare un ordine di grandezza (compreso tra 10 e 10^2 ore, dato che l'andamento reale risulta più lento di quello lineare) utile a stimare l'intervallo temporale entro il quale la produzione di calore risulta sicuramente efficace a contrastare la dispersione. Trascosto un tempo maggiore la produzione di calore potrebbe risultare non sufficiente conducendo la colonia al collasso termico.



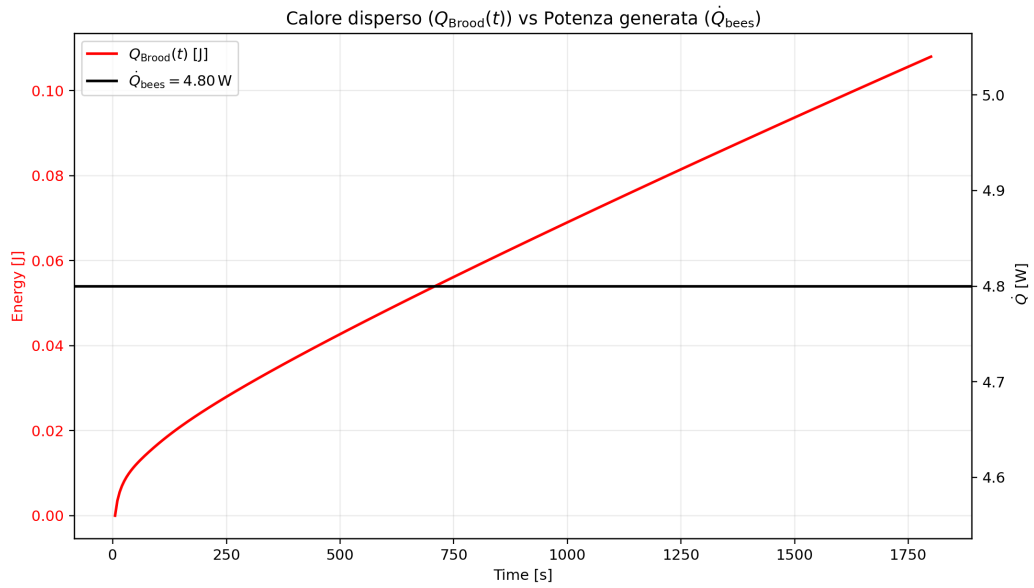
(a) Cassetta doppia



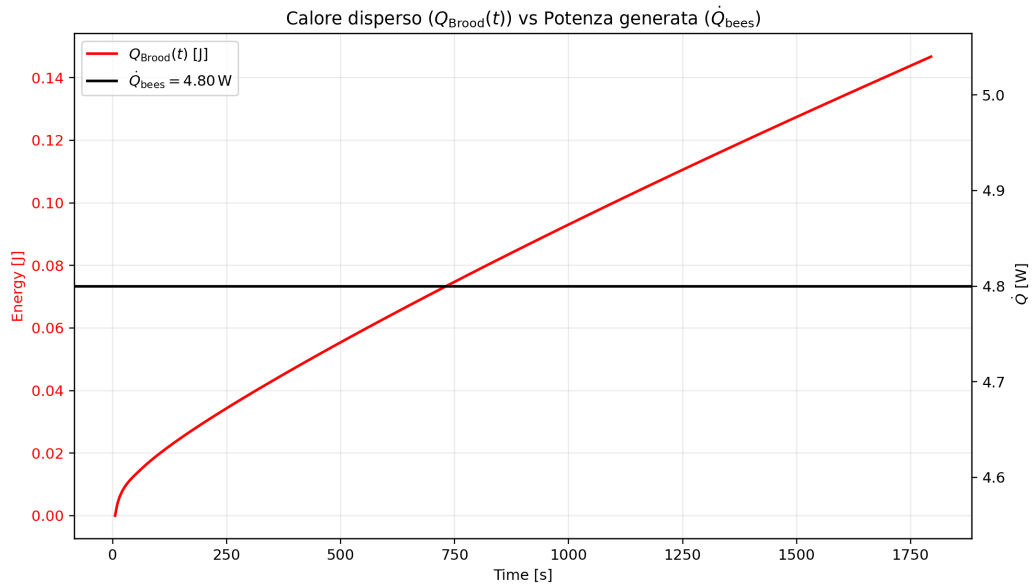
(b) Cassetta singola

Figura 5.6: Calore disperso (rosso) e calore generato (nero) con modello a potenza costante

Questa analisi in particolare si pone come punto di partenza per l'elaborazione di un modello più accurato e puntuale; l'adozione del modello in questione ha condotto a risultati decisamente privi di significato fisico, differenze tali in termini di ordini di grandezza evidenziano la necessità di approfondire la comprensione del fenomeno per elaborare un modello attendibile.



(a) Cassetta doppia



(b) Cassetta singola

Figura 5.7: Calore disperso (rosso) e potenza generata (nero) con modello a potenza costante, i grafici non si intersecano se si osservano le scale, tuttavia è stato necessario porli in questo modo per motivi di rappresentazione

Conclusioni

Il presente lavoro di tesi ha avuto l'obiettivo di impiegare modellazioni CFD-CHT per studiare lo scambio termico all'interno di un alveare, in un caso bidimensionale semplificato. Le simulazioni mostrano che in alveari esposti per periodi prolungati (dell'ordine dei giorni) a condizioni climatiche critiche i meccanismi di termoregolazione per evitare che la temperatura scenda al di sotto della soglia critica di 21 °C potrebbero non essere sufficienti.

È emerso, inoltre, che uno sviluppo verticale dell'arnia favorisce la conservazione del calore, riducendo il fabbisogno di produzione termica sino a circa il 30%. Tuttavia, il perdurare di condizioni severe compromette la stabilità e la salute della colonia, anche a causa dello stress metabolico cui essa è sottoposta, con potenziali ripercussioni sulla sopravvivenza.

Questo studio si propone come punto di partenza per ricerche più strutturate e mirate, volte ad analizzare in maggior dettaglio i meccanismi di scambio termico. Un approccio numerico-computazionale potrà contribuire a individuare le criticità fondamentali e, in prospettiva, a definire strategie utili a ridurre la mortalità delle api in ambienti sfavorevoli.

Appendice A

Governing equations

[11] Equazioni di Navier-Stokes

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (\text{A.1})$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\tau} = \rho \mathbf{g}, \quad (\text{A.2})$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{u}) + \nabla \cdot (p \mathbf{u}) - \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u}) - \nabla \cdot (k \nabla T) = \rho r + \rho \mathbf{g} \cdot \mathbf{u}. \quad (\text{A.3})$$

- ρ densità del fluido [kg/m³]
- $\mathbf{u}$ vettore velocità [m/s]
- p pressione [Pa]
- $\boldsymbol{\tau}$ tensore degli sforzi viscosi [Pa]
- $\mathbf{g}$ accelerazione di gravità [m/s²]
- E energia totale per unità di massa [J/kg]
- T temperatura [K]
- k conducibilità termica [W/(m · K)]
- r sorgente termica per unità di massa [W/kg]

[11] Equazione della conduzione termica nei solidi

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + Q$$

- ρ densità del materiale [kg/m³]
- c_p calore specifico a pressione costante [J/(kg · K)]
- T temperatura [K]
- t tempo [s]

- k conducibilità termica $[\text{W}/(\text{m} \cdot \text{K})]$
- Q sorgente di calore volumetrica $[\text{W}/\text{m}^3]$

Appendice B

Tutorial

This paragraph aims to provide a practical guide, enriched with the underlying theoretical motivations, for the creation and management of a coupled simulation case. The intent is to offer a useful and replicable tool to those who wish to deepen the study, providing a clear and modifiable reference structure.

Case structure

The general structure of the case follows the typical one of an **OpenFOAM** simulation, with the addition of the directories and files required for coupling for the **CHT** exchange. In particular, the overall layout takes the following form:

```
Case
  Fluid
    0
      alphasat
      k
      nut
      omega
      p
      p_rgh
      T
      U
    constant
      g
      thermophysicalProperties
      transportProperties
      turbulenceProperties
    system
      blockMeshDict
      controlDict
      decomposeParDict
      fvSchemes
      fvSolution
      preciceDict
```

```
Solid
  Solid.inp
  config.yml
precice-config.xml
```

The initialized and ready-to-run case can be downloaded from the GitHub repository, complete with `.sh` files for automation and Python scripts for post-processing, whose function has been described in the central part of the thesis work. <https://github.com/mauriziocere/HiveCHT/tree/bc5a162d9182c167a5aadd67-0bc201a3d5c6e5b1/HiveCHT>

Mesh generation

For the generation of the fluid mesh, it is sufficient to move via terminal into the directory `Case/Fluid` and execute the command:

```
blockMesh
```

If you wish to manually modify the cell size, for example to intervene on the `CO` value, you can act directly on the `blocks` section of the `blockMeshDict` file, making sure to maintain continuity in the number of cells between adjacent blocks.

As for the solid, instead, the mesh is not directly modifiable from the `Solid.inp` file. It is necessary to remodel the geometry using `FreeCAD`, export it in `.step` format, import it into `prePoMax`, and perform a new discretization.

Case adaptation

The flexibility of the setup allows the exploration of different experimental and numerical variants.

- **Fluid side:** The possibilities for modification are numerous. It is possible, for example, to:
 - vary the boundary conditions, replacing for example a wall with a `symmetryPlane` condition;
 - reduce the inlet velocities to analyze the case in laminar regime, where natural convection dominates;
 - intervene on the thermophysical properties (`thermophysicalProperties`), studying the effect of different fluids or operating conditions.
- **Solid side:** The possibilities of intervention mainly concern:
 - the modification of material properties, in order to evaluate the impact of conductivity or heat capacity;
 - the variation of calculation parameters in the simulation step and/or the obtainable outputs.

In summary, the architecture of the case provides a robust and at the same time versatile framework: on the one hand, it guarantees the reproducibility of the presented results, on the other, it allows wide possibilities of customization depending on the study objectives.

Coupling setup in preCICE

Greater relevance will be given to the changes that can be implemented in the coupling, since they are of primary interest for a CHT case study and atypical compared to a CFD simulation with OpenFOAM. For this purpose, the structure of `precice.config.xml` will be highlighted in order to justify the choices made and to provide a theoretical basis for possible modifications.

Participants and meshes. Two participants have been defined: `Fluid` and `Solid`. The fluid registers as mesh the faces of the wall exposed to heat transfer, while the solid registers two entities: a nodal mesh to read the temperatures and a surface mesh to receive the fluxes (in accordance with what is declared respectively in `preciceDict` and `config.yml`). This separation is necessary with the selected geometric mapping because in OpenFOAM the exchange naturally occurs at face level, while in CalculiX the primary thermal quantities are related to the nodes (for possible changes to the geometric mapping and the associated required meshes see [11]). The duplication of the mesh on the solid side therefore ensures compatibility between the two discretizations.

```
<mesh name="Fluid-Mesh" dimensions="2">
  <use-data name="Temperature" />
  <use-data name="Heat-Flux" />
</mesh>

<mesh name="Solid-Mesh" dimensions="2">
  <use-data name="Heat-Flux" />
</mesh>

<mesh name="Solid-Mesh-nodes" dimensions="2">
  <use-data name="Temperature" />
</mesh>
```

Exchanged data. The quantities considered for a purely CHT case are:

- **Temperature:** transmitted in this case from the fluid to the solid, so that the solid domain can impose updated thermal values at the boundary at each time window.
- **Heat-Flux:** returned from the solid to the fluid, so that `buoyantPimpleFoam` receives the correct thermal gradients to be used as temperature boundary conditions.

This choice is almost arbitrary: the fluid imposes the temperature conditions at the interface, while the solid reacts by returning the resulting heat flux; however, the inverse setup would also be possible.

```
<exchange data="Temperature" mesh="Fluid-Mesh" from="Fluid" to="Solid" />
<exchange data="Heat-Flux" mesh="Solid-Mesh" from="Solid" to="Fluid" />
```

Geometric mapping. A nearest-neighbor mapping with `consistent` constraint was chosen. The nearest-neighbor ensures a direct and stable correspondence, avoiding spurious interpolations. The choice was mainly motivated by the fact that the solid was modeled as a single body (single heat transfer surface between Frame+Brood and Fluid); therefore, by adopting this type of mapping, there is no risk of cells belonging to the frame interacting with fluid cells near the brood and vice versa.

```
<mapping:nearest-neighbor
  direction="read"
  from="Solid-Mesh"
  to="Fluid-Mesh"
  constraint="consistent" />
<mapping:nearest-neighbor
  direction="read"
  from="Fluid-Mesh"
  to="Solid-Mesh-nodes"
  constraint="consistent" />
```

Coupling scheme. The chosen scheme is of type `serial-implicit`, with order $\text{Fluid} \rightarrow \text{Solid}$. This means that, within each time window, the two domains exchange data iteratively until convergence is reached. The choice of an implicit scheme, rather than an explicit one, was motivated by the strong thermo-fluid coupling: the flux gradients and temperatures mutually affect numerical stability. The implicit approach allows the use of larger time steps without compromising the robustness of the solution.

```
<coupling-scheme:serial-implicit>
  <time-window-size value="0.1" />
  <max-time value="30" />
  <min-iterations value="2" />
  <max-iterations value="40" />
  <participants first="Fluid" second="Solid" />
  ...
</coupling-scheme:serial-implicit>
```

Convergence criterion. Convergence is evaluated through the relative 2-norm of the difference between two successive iterations:

$$r^{(k)} = \frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2}{\|\mathbf{x}^{(k)}\|_2},$$

where $\mathbf{x}^{(k)}$ is the vector of the values exchanged at iteration k . The denominator normalizes with respect to the field scale, so that large absolute values do not mask still significant variations. For example, `limit=1e-3` indicates that convergence is considered reached when the relative variation is below 0.1%.

```
<relative-convergence-measure limit="2.0e-3" data="Temperature" mesh="Fluid-Mesh" />
<relative-convergence-measure data="Heat-Flux" mesh="Solid-Mesh" limit="3.0e-3"/>
```

Numerical accelerator. To speed up convergence, the IQN-ILS (Interface Quasi-Newton with Inverse Least Squares) accelerator was employed. This method builds an approximation of the inverse Jacobian matrix exclusively from the input/output vectors stored in the previous iterations. The quality and the amount of such information are crucial for the speed and robustness of convergence [? ?].

In practice, the main parameters are:

- **initial-relaxation:** controls the initial phase, allowing for a more cautious progression (0.1–0.25);
- **max-used-iterations:** limits the memory of the scheme, maintaining a balance between accuracy and computational cost.

Other acceleration strategies (Aitken, IQN-IMVJ) has generally showed a lower efficiency.

```
<acceleration:IQN-ILS>
  <data mesh="Solid-Mesh" name="Heat-Flux" />
  <initial-relaxation value="0.25" />
  <max-used-iterations value="15"/>
</acceleration:IQN-ILS>
```

For further information on these topics, please refer to the preCICE documentation.

Case execution

The execution of the simulation requires few and simple commands, designed to also facilitate some phases of post-processing. For simplicity, it was preferred not to provide a `run.sh` script, but to proceed directly by opening two distinct terminals in the directories `Case/Fluid` and `Case/Solid`.

Fluid domain

After generating the mesh with `blockMesh`, it is necessary to execute:

```
decomposePar
mpirun -np 4 buoyantPimpleFoam -parallel > solver.log 2>&1
```

In this way, 4 processors are used, with a consequent reduction of computation times and, moreover, all the information on the fluid run is collected in `solver.log`, which can be read by `CO.py` and allows to keep track of the computational part at each *time-step*.

Solid domain

In the **Solid** folder, instead, execute:

```
ccx_preCICE -i Solid -precice-participant Solid
```

(or the corresponding command, if the adapter was saved with a different name).

The simulation duration is controlled by the **precice-config.xml** file and should be consistent with the parameters defined in **controlDict** and with the time step of the solid domain. It is essential to ensure that the number of increments satisfies the condition:

$$\text{INC} \geq \frac{\text{Simulation time}}{\Delta t_{\text{simulation}}}.$$

Post-processing and Cleanup

At the end of the run, the fluid domain is reconstructed with:

```
reconstructPar
```

Subsequently, it is recommended to clean the spurious folders, generated for example by *functionObjects* (folders such as 0.9, 1.9, etc.). For this purpose, the script **adjust.sh** has been prepared, which deletes the spurious folders not constituted by integers.

Two support scripts **clean.sh** are also available in the **Fluid** and **Solid** directories, which quickly delete non-essential files and folders, keeping only the fundamental data and the necessary scripts in case you want to restart the case without manual deletion of the folders.

Finally, the visualization of the simulation can be performed via **ParaView**. For this purpose, a Python script has been inserted for the conversion of the data associated with the solid into **.vtk** format readable by **ParaView**. For correct use of the script, it is necessary to move the **.frd** file into the **Case/Solid/ccxToParaview** directory and run the script by entering the command from the terminal:

```
python3 ccx2paraview.py Solid.frd vtk
```

For post-processing, the **.py** codes used and described in the dedicated chapter have been made available, all runnable as:

```
python3 script_name.py
```

except for **CO.py** which requires the log from which to extract the data, therefore:

```
python3 CO.py --log solver.log --out CO.png
```

Bibliografia

- [1] E. Minaud, F. Rebaudo, G. Mainardi, P. Vardakas, F. Hatjina, I. Steffan-Dewenter, F. Requier, Temperature in overwintering honey bee colonies reveals brood status and predicts colony mortality, *Ecological Indicators*, 2024.
- [2] D.M. Mitchell, Are man-made hives valid thermal surrogates for natural honey bee nests (*Apis mellifera*)?, *Journal of Thermal Biology*, 2024.
- [3] D.M. Mitchell, Honey bee (*Apis mellifera*) size determines colony heat transfer when brood covering or distributed, *International Journal of Biometeorology*, 2022.
- [4] S. Rodríguez-Vásquez, A.A. Gardea-Béjar, A. Romo-Chacón, J. García-Hernández, H. González-Ríos, J. Antonio Orozco-Avitia, Hive temperature regulation by honeybees in response to extreme conditions, *Agrociencia*, 2024.
- [5] D.M. Mitchell, Ratios of colony mass to thermal conductance of tree and man-made nest enclosures of *Apis mellifera*: implications for survival, clustering, humidity regulation and *Varroa destructor*, *International Journal of Biometeorology*, 2016.
- [6] L. Fahrenholz, I. Lamprecht, B. Schricker, Thermal investigations of a honey bee colony: thermoregulation of the hive during summer and winter and heat production of members of different bee castes, *Journal of Comparative Physiology B*, 1989.
- [7] H. Jarimi, E. Tapia-Brito, S. Riffat, A Review on Thermoregulation Techniques in Honey Bees' (*Apis Mellifera*) Beehive Microclimate and Its Similarities to the Heating and Cooling Management in Buildings, *Future Cities and Environment*, 2020.
- [8] A. Stabentheiner, H. Kovac, M. Mandl, H. Käfer, Coping with the cold and fighting the heat: thermal homeostasis of a superorganism, the honeybee colony, *Journal of Comparative Physiology A*, 2021.
- [9] J.A.C. Humphrey, E.S. Dykes, Thermal energy conduction in a honey bee comb due to cell-heating bees, *Journal of Theoretical Biology*, 2008.
- [10] R. Sudarsana, C. Thompson, P.G. Kevan, H.J. Eberla, Flow currents and ventilation in Langstroth beehives due to brood thermoregulation efforts of honeybees, *Journal of Theoretical Biology*, 2012.

-
- [11] L. Cheung Yau, Conjugate Heat Transfer with the multiphysics coupling library preCICE, in: Master's Thesis, 2016
 - [12] B.K.M. Saleem, A. Mustafa, D.A. Kareem, M.I. Yuce, M. Szydłowski, N. Al-Ansari, Numerical Analysis of Turbulent Flow over a Backward-facing Step in an Open Channel, *Hydro-Engineering and Environmental Mechanics*, Vol. 70, No. 1, pp. 41–52, 2023.
 - [13] A.L. Destri, Analisi di compositi laminati a fibra curva tramite ABAQUS, in: Tesi di laurea, 2023/2024, pp 33; <https://webthesis.biblio.polito.it/34297/1/tesi.pdf>
 - [14] S.S. Rudi, *Compendio Tabelle Termofisiche*, 2015
 - [15] J.E.J. Staggs, Thermal conductivity estimates of intumescent chars by direct numerical simulation, *Fire Safety Journal*, Vol. 45, Issue 4, pp. 228–237, June 2010.
 - [16] Y. Jannot, A. Degiovanni, V. Schick, J. Meulemans, Thermal diffusivity measurement of insulating materials at high temperature with a four-layer (4L) method, *International Journal of Thermal Sciences*, Vol. 150, 2020.