

POLITECNICO DI TORINO

---

Collegio di Ingegneria Chimica e dei Materiali

Corso di Laurea Magistrale in Ingegneria Chimica e dei Processi  
Sostenibili

Tesi di Laurea Magistrale

**Implementazione di modelli *data-driven*  
per applicazioni cromatografiche basati  
su simulazioni fluidodinamiche**



**Relatore**

Prof. Daniele MARCHISIO

**Correlatori:**

Prof. Gianluca BOCCARDO

Prof. Antonio BUFFO

**Studente**

Andrea STRAZZOLINI  
matricola: 263929

**Supervisore aziendale**

GSK Vaccines Srl

Dott. Ing. Antonio Gaetano CARDILLO

---

ANNO ACCADEMICO 2020 – 2021



# Indice

Elenco delle figure	v
Elenco delle tabelle	vii
<b>1 Introduzione</b>	<b>1</b>
<b>2 La modellazione fluidodinamica in ambito cromatografico</b>	<b>5</b>
2.1 La cromatografia nell'industria farmaceutica . . . . .	5
2.2 Nozioni fondamentali di fluidodinamica computazionale . . . . .	9
2.2.1 Equazioni di governo . . . . .	9
2.2.2 Metodi numerici . . . . .	11
2.2.3 Griglie di calcolo . . . . .	16
2.3 Modellazione di colonne cromatografiche . . . . .	17
2.3.1 Modellazione alla scala della colonna . . . . .	19
2.3.2 Modellazione alla scala delle particelle . . . . .	19
2.3.3 Modellazione CFD dello scambio termico all'interno di letti impac-	
cati: stato dell'arte . . . . .	20
<b>3 I modelli <i>data-driven</i> e le loro applicazioni nell'ambito dell'industria di</b>	<b>23</b>
<b>processo</b>	
3.1 Le reti neurali . . . . .	24
<b>4 Impostazione delle simulazioni e costruzione del modello</b>	<b>29</b>
4.1 Costruzione della geometria . . . . .	29
4.1.1 Geometria del contenitore . . . . .	32
4.1.2 Disposizione iniziale delle sfere . . . . .	32
4.1.3 Distribuzione dei diametri delle sfere . . . . .	33
4.1.4 Condizioni di simulazione . . . . .	34
4.1.5 Modellazione dei punti di contatto . . . . .	35
4.2 Impostazione delle simulazioni fluidodinamiche . . . . .	37
4.2.1 Geometria e griglia di calcolo . . . . .	38
4.2.2 Proprietà dei materiali e condizioni al contorno . . . . .	38
4.2.3 Schemi numerici e criteri di convergenza adottati . . . . .	41
4.3 Progettazione del <i>dataset</i> . . . . .	41
4.4 Analisi preliminari . . . . .	44
4.4.1 Selezione del volume di controllo . . . . .	44
4.4.2 Discretizzazione spaziale e temporale . . . . .	45
4.4.3 Analisi della varianza . . . . .	49
4.4.4 Calcolo del coefficiente di scambio termico . . . . .	50
4.5 Costruzione della rete neurale . . . . .	54

<b>5</b>	<b>Discussione dei risultati</b>	57
5.1	Rete predittiva dei parametri geometrici . . . . .	57
5.2	Rete predittiva del coefficiente di scambio termico . . . . .	60
<b>6</b>	<b>Conclusioni e prospettive future</b>	67
	<b>Bibliografia</b>	69
<b>A</b>	<b><i>Script</i> per la creazione della geometria in Blender</b>	73
<b>B</b>	<b><i>Script</i> per la modifica dei punti di contatto in Salome</b>	77
	<b>Lista dei simboli</b>	81
	<b>Acronimi</b>	85
	<b>Ringraziamenti</b>	87

## Elenco delle figure

2.1	Schematizzazione del processo produttivo del vaccino ricombinante contro l'epatite B (Recombivax HB, Merck) [6]. . . . .	8
2.2	Schematizzazione del processo produttivo di un generico anticorpo monoclonale [6]. . . . .	9
2.3	<i>Workflow</i> per la ricerca e l'individuazione di ligandi per la cromatografia di affinità [13]. . . . .	10
2.4	Rappresentazione schematica dei meccanismi di trasporto presenti all'interno di un riempimento non strutturato di particelle di resina cromatografica [13]. . . . .	10
2.5	Esempio di celle tridimensionali esaedriche. . . . .	14
2.6	Schematizzazione della procedura di <i>meshing</i> della utility <code>snappyHexMesh</code> [18]. . . . .	17
2.7	Schematizzazione dei metodi di modifica globale (casi a, b) e locale (casi c, d) dei punti di contatto: (a) <i>Gaps</i> ; (b) <i>Overlaps</i> ; (c) <i>Bridges</i> ; (d) <i>Caps</i> [35]. . . . .	21
3.1	Schema di un neurone artificiale. . . . .	25
3.2	Rappresentazione schematica di un perceptrone, in dettaglio le operazioni compiute da un singolo neurone. . . . .	26
3.3	Rappresentazione schematica di una rete neurale profonda [42]. I nodi in basso, rappresentati con "+1", rappresentano il <i>bias</i> di ciascun <i>layer</i> . . . . .	26
3.4	Principali funzioni di attivazione. . . . .	28
3.5	Rappresentazione grafica dell'effetto del <i>learning rate</i> nell'ottimizzazione di una funzione del tipo $z = x^2 + ay^2$ , con $a > 1$ [44]. Ogni vettore che compone la spezzata rappresenta un'iterazione. . . . .	28
4.1	Vista frontale e dall'alto del cilindro di contenimento delle sfere, del piano sottostante e del tronco di cono sovrastante. . . . .	32
4.2	Distanze minime considerate tra le pareti del contenitore e il volume estratto per la simulazione. . . . .	33
4.3	Rappresentazione schematica della griglia di disposizione iniziale delle sfere. . . . .	33
4.4	Esempio di distribuzione dimensionale delle sfere generata all'interno della simulazione, avente diametro medio pari a $40 \mu m$ e deviazione standard pari a $0.08 \times 10^6 \mu m^3$ . Le curve riportate in rosso sono rispettivamente la funzione densità di probabilità e la sua radice cubica. . . . .	34
4.5	Schematizzazione della procedura di implementazione del <i>bridges method</i> . . . . .	36
4.6	Esempio di una geometria esportata in formato STL in seguito al trattamento dei punti di contatto. . . . .	37
4.7	Esempio dei campi di velocità e pressione calcolati lungo una sezione parallela all'asse z. . . . .	39

4.8	Rappresentazione schematica del dominio di simulazione. . . . .	40
4.9	Esempio di dominio fluido simulato. Viene riportata la relativa suddivisione delle facce del parallelepipedo. . . . .	40
4.10	Esempio di dominio resina simulato. Viene riportata la relativa suddivisione delle facce del parallelepipedo. . . . .	41
4.11	Confronto tra le distribuzioni di massima ampiezza generate attraverso l'Eq. (4.14) e gli intervalli di diametri forniti in letteratura (in rosso) per determinati valori di diametro medio. . . . .	43
4.12	Porosità esterne calcolate nei diversi casi considerati in Tab. 4.5. . . . .	46
4.13	Numero di sfere incluse e numero di celle costruite all'interno del volume di simulazione. . . . .	46
4.14	Valori calcolati di porosità esterna, area interfacciale specifica, coefficiente di Darcy e coefficiente di scambio termico a diverse dimensioni di cella base. . . . .	48
4.15	Particolare di una sezione della griglia di calcolo per diversi valori di cella base. Le dimensioni di cella sono riportate in Tab. 4.6. . . . .	48
4.16	Istante alla quale la temperatura media nel dominio resina ( $\bar{T}_s$ ) è ridotta di un Kelvin rispetto alla temperatura iniziale. . . . .	49
4.17	Profilo di temperatura lungo una sezione longitudinale e trasversale del dominio di simulazione a $t = 1$ s. La temperatura è espressa in Kelvin. . . . .	52
4.18	Esempio di evoluzione del profilo di temperatura nel dominio fluido (a sinistra) e nel dominio resina (a destra) a $t = 0, 1, 2$ e $4$ s. . . . .	53
5.1	Esempio dell'andamento della funzione di perdita, l'errore assoluto medio, su <i>training</i> e <i>validation set</i> in caso di <i>underfitting</i> e <i>overfitting</i> dei dati. . . . .	57
5.2	Andamento dell'errore assoluto medio (MAE) e dello scarto quadratico medio (MSE) nel corso dell'allenamento della rete per la predizione dei parametri geometrici. . . . .	59
5.3	Diagrammi di parità delle predizioni effettuate sul <i>test set</i> . . . . .	63
5.4	Distribuzioni degli errori assoluti e relativi sulle predizioni effettuate dalla rete per la predizione di porosità esterna ( $\epsilon$ ), area interfacciale specifica ( $a_{fs}$ ) e coefficiente di Darcy ( $d$ ). . . . .	64
5.5	Andamento dell'errore assoluto medio (MAE) e dello scarto quadratico medio (MSE) nel corso dell'allenamento della rete per la predizione del coefficiente di scambio termico. . . . .	64
5.6	Diagramma di parità delle predizioni del coefficiente di scambio termico effettuate sul <i>test set</i> . . . . .	65
5.7	Distribuzioni degli errori assoluti e relativi sulle predizioni effettuate dalla rete per la predizione del coefficiente di scambio termico. . . . .	65

## Elenco delle tabelle

4.1	Impostazioni utilizzate per la simulazione in <code>Blender</code> . . . . .	35
4.2	Proprietà termofisiche dei materiali. . . . .	38
4.3	Fattori di sottorilassamento e criteri di convergenza adottati. . . . .	42
4.4	Intervalli di variazione delle <i>features</i> nella costruzione del <i>dataset</i> . . . . .	42
4.5	Volumi considerati nella selezione del REV. Le dimensioni del parallelepipedo sono espresse in multipli del diametro medio. . . . .	45
4.6	Dimensioni di cella base considerate nel corso dell'analisi di <i>grid independence</i> , espresse in frazioni del diametro medio delle sfere. . . . .	47
4.7	Parametri monitorati nel corso dell'analisi di <i>grid independence</i> . . . . .	47
4.8	Discretizzazione temporale utilizzata nei diversi casi considerati nell'analisi di indipendenza. . . . .	49
4.9	Quantità calcolate, valore medio e deviazione massima dal valor medio ottenuti nel corso dell'analisi della varianza. . . . .	50
4.10	Deviazione massima dal valor medio del coefficiente di scambio termico osservata calcolandolo con diversi $\Delta T$ di riferimento. . . . .	52
5.1	Combinazioni di iperparametri testate e rispettivi errori relativi medi sulle quantità predette. Il numero di elementi in parentesi quadre rappresenta il numero di <i>hidden layer</i> , mentre il valore rappresenta il numero di neuroni per ciascun <i>layer</i> . . . . .	59
5.2	Errori relativi sulle predizioni di porosità esterna ( $\epsilon$ ), area interfacciale specifica ( $a_{fs}$ ) e coefficiente di Darcy ( $d$ ) al variare della dimensione del <i>dataset</i> di allenamento. . . . .	60
5.3	Varianza calcolata e errore relativo medio sulle predizioni di porosità esterna, area interfacciale specifica e coefficiente di Darcy. . . . .	60
5.4	Combinazioni di iperparametri testate e rispettivi errori assoluti e relativi medi sulle predizioni. Il numero di elementi in parentesi quadre rappresenta il numero di <i>hidden layer</i> , mentre il valore rappresenta il numero di neuroni per ciascun <i>layer</i> . . . . .	61
5.5	Errori relativi sulle predizioni della rete al variare della dimensione del <i>dataset</i> di allenamento. . . . .	62
5.6	Varianza calcolata e errore relativo medio sulle predizioni del coefficiente di scambio termico. . . . .	62

## Riassunto

La previsione della capacità di separazione e la definizione delle condizioni operative di un'apparecchiatura cromatografica trovano attualmente fondamento in una forte base empirica. In un'ottica di ottenimento di modelli ai principi primi o semi-empirici con validità più generale, si rende necessario affrontare gli elementi critici della descrizione dei fenomeni di trasporto che si verificano all'interno di una colonna cromatografica, legati in particolar modo alle scale dimensionali estremamente diverse alla quale avvengono questi fenomeni. Particolare attenzione viene qui posta ai fenomeni che avvengono alle piccole scale, cioè gli interstizi fra le particelle del riempimento della colonna cromatografica (in inglese *particle-scale*), e quelli che avvengono alle grandi scale, cioè la colonna stessa (in inglese *column-scale*).

In questa Tesi viene proposta una metodologia innovativa per il calcolo di alcune proprietà di interesse (quali ad esempio porosità esterna, area interfacciale specifica, permeabilità e coefficiente di scambio termico) relative al *bulk* di letti impaccati non strutturati di colonne cromatografiche, attraverso la costruzione di un modello predittivo basato su reti neurali. In particolare, viene proposto un *workflow* per la generazione e l'elaborazione *in silico* di geometrie assimilabili a porzioni di letto impaccato non strutturato di particelle sferiche, le quali sono impiegate in simulazioni CFD necessarie alla costruzione del *dataset* alla base del modello predittivo. I risultati ottenuti indicano che è possibile costruire reti neurali con sufficiente accuratezza, specialmente per quanto riguarda quantità come la porosità esterna del letto e l'area interfacciale specifica, anche utilizzando insiemi relativamente modesti di dati. Vengono inoltre delineate alcune strategie per un futuro perfezionamento del *workflow* qui proposto.

Il presente lavoro è stato realizzato in collaborazione con il centro ricerche di GSK Vaccines S.r.l. (Siena, SI), azienda operante nella produzione di prodotti vaccinali per il contrasto delle malattie infettive.

# 1. Introduzione

Il concetto di *Digital Twin* (DT) è stato introdotto nel 2003 da Michael Grieves, con esso si intende una rappresentazione virtuale di un sistema reale avente lo scopo di ridurre il *gap* tra la progettazione di un processo e la sua effettiva realizzazione fisica [1]. I costituenti fondamentali di un modello *digital twin* sono: il sistema fisico in esame, insieme a tutto ciò che permette l'estrazione di dati da esso, il sistema virtuale in grado di rappresentarne il comportamento e infine le connessioni e i flussi di informazioni che permettono di allineare il processo fisico e quello virtuale. Nel corso dei due decenni appena trascorsi sono cresciute esponenzialmente sia la quantità e la disponibilità di informazioni che la capacità di analisi ed elaborazione di queste, fornendo le basi per la crescita e lo sviluppo di tale ambito della modellistica. Nello scorso decennio, questo concetto è stato progressivamente introdotto anche nel mondo dell'industria, definito come "una simulazione multifisica, multiscala, probabilistica, a elevata fedeltà che riflette in tempo reale lo stato di un corrispondente fisico sulla base di dati storici, sensori connessi in tempo reale e modelli fisici" [2]. Un crescente numero di ambiti e potenzialità di applicazione dei DT vengono individuati, spaziando dall'ottimizzazione del design di prodotto, all'ingegnerizzazione delle singole operazioni e degli interi processi produttivi fino alla gestione del ciclo di vita dei prodotti e degli impianti produttivi [3]. I DT sono stati applicati con successo, da parte di aziende leader del settore, in settori quali l'ingegneria aerospaziale, le reti elettriche, le energie rinnovabili, la gestione dei rifiuti, il petrolchimico, l'assistenza sanitaria.

La modellazione fisica e la simulazione sono le basi dell'implementazione pratica dei *digital twins* nell'industria di processo, rappresentano infatti l'elemento chiave di collegamento tra il modello digitale e il flusso di informazioni proveniente dal corrispondente fisico. Inoltre, laddove non fossero disponibili dati empirici sufficienti, esse permettono di generare nel mondo virtuale risposte tempestive e accurate che possono essere utilizzate per effettuare delle scelte in condizioni di incertezza. Da qui nasce l'interesse nel costruire modelli meccanicistici e probabilistici che siano sempre più affidabili e veloci, e che possano essere integrati con tecniche di intelligenza artificiale per l'elaborazione automatica delle informazioni e delle risposte.

La fluidodinamica computazionale (CFD, *Computational Fluid Dynamics*) è una tecnica ad oggi ben affermata nell'ambito dell'ingegneria chimica, ed è ampiamente utilizzata per la previsione delle prestazioni di un'apparecchiatura e per la descrizione delle caratteristiche del moto e del trasporto di determinate quantità, come il calore o le specie chimiche, all'interno di essa. La CFD si basa sulla risoluzione numerica delle equazioni di bilancio di quantità di moto, energia e materia, e permette di simulare fenomeni anche di complessità elevata come flussi multifase, reazioni chimiche e cambi di fase. Essa rappresenta quindi uno strumento rapido e poco dispendioso per ricavare informazioni sufficientemente accurate da essere impiegate per prendere decisioni nella progettazione o nella definizione delle condizioni operative di un'apparecchiatura o di un impianto. Inoltre, nell'ambito del *design* di nuovi dispositivi o nello *scaling-up* di prototipi, permette di

estendere il campo di indagine al di fuori delle condizioni osservate empiricamente, una volta eseguita la calibrazione dei modelli fisici impiegati. Quest'ultima considerazione, unita alla possibilità di generare grandi moli di dati *in silico* anche in maniera automatica, la rende un campo di applicazione ideale di metodologie di *machine learning*. Il *machine learning* (ML) è un ambito dell'intelligenza artificiale che si occupa della costruzione di algoritmi in grado di "imparare" a svolgere un determinato compito attraverso i dati forniti, ovvero sia essere in grado di adattare se stessi ai nuovi dati che gli vengono sottoposti, migliorando progressivamente le proprie prestazioni. Per loro natura, le tecniche di ML offrono grandi vantaggi quando utilizzano grandi insiemi non strutturati di informazioni, e sono in grado di evolversi in modo dinamico tramite l'acquisizione continua di nuovi dati. Di conseguenza, procedure automatizzate di simulazione, raccolta ed elaborazione di questi dati, specialmente se eseguite direttamente attraverso strumenti informatici, permettono la costruzione di modelli predittivi con un ridotto impiego di tempo e risorse e in grado di fornire risposte in modo quasi istantaneo.

Nel campo della chimica fine, e in particolare nei processi biotecnologici tipici dell'industria farmaceutica, la cromatografia riveste un ruolo chiave nell'ottenimento del prodotto finito. Essa è una tecnica di separazione ampiamente impiegata nella concentrazione e nella purificazione di prodotti quali le macromolecole organiche. Esistono un gran numero di tecniche cromatografiche differenti, le quali prevedono generalmente interazioni complesse tra la soluzione da risolvere e i componenti dell'apparecchiatura. Per questo motivo, non esistono dei modelli generalizzati per la stima delle prestazioni e il *design* è basato in larga parte su prove sperimentali. Ciò rappresenta un limite considerevole nella costruzione di *digital twins* di queste apparecchiature, e in ambiti industriali si registra un crescente sforzo per superare questo ostacolo. In questo senso, le possibilità offerte dalla CFD sono molteplici, grazie alla sua capacità di simulare fenomeni su scale anche molto diverse tra loro e di integrare le informazioni ricavate secondo un approccio *bottom-up*, muovendosi dal livello di dettaglio maggiore sino alla scala dell'intera apparecchiatura. L'integrazione di modelli CFD e ML rappresenta un ulteriore passo avanti, poiché è in grado di ridurre sensibilmente i tempi di risposta del DT, eliminando la necessità di eseguire simulazioni CFD in tempo reale e rendendo di fatto quest'ultimo un modello dinamico.

Il presente lavoro, realizzato in collaborazione con GSK Vaccines S.r.l., propone una metodologia innovativa per la predizione della porosità esterna, dell'area interfacciale specifica, della permeabilità e del coefficiente di scambio termico in letti impaccati non strutturati di colonne cromatografiche deputate alla separazione di prodotti proteici da soluzioni acquose. In particolare, le quantità di interesse individuate sono state calcolate attraverso simulazioni CFD alla scala delle particelle di resina cromatografica, individuando un elemento di volume in grado di fornire risultati validi per l'intero *bulk* del letto impaccato. La procedura di simulazione è stata applicata per la costruzione di un *dataset*, impiegato successivamente per la costruzione di un modello predittivo basato su *shallow deep neural networks*, per poi valutarne le prestazioni. Molte delle considerazioni ricavate per quanto riguarda lo scambio termico sono valide anche in problemi di maggior interesse applicativo, quale lo scambio di materia, e la trattazione è stata ideata

per poter essere integrata in simulazioni dello scambio termico alla scala dell'intera colonna. Il lavoro si inserisce in un contesto più ampio relativo alla costruzione di *digital twins* di apparecchiature cromatografiche, il quale valuta e integra numerose metodologie differenti.

Il presente documento è strutturato come segue:

- presentazione di elementi teorici riguardanti la cromatografia e la fluidodinamica computazionale, unita alla descrizione dello stato dell'arte dell'applicazione della CFD a problemi riguardanti letti impaccati non strutturati e delle sue potenzialità nell'ambito della cromatografia (Capitolo 2);
- presentazione dei concetti fondamentali dei metodi di *machine learning*, e in particolare delle reti neurali profonde (Capitolo 3);
- descrizione delle procedure impiegate nelle varie fasi del lavoro, in particolare nella costruzione ed elaborazione della geometria, nell'impostazione delle simulazioni CFD, nella costruzione del *dataset* e delle reti neurali alla base del modello predittivo (Capitolo 4);
- presentazione dei risultati ottenuti e relativa analisi (Capitolo 5);
- presentazione delle considerazioni conclusive (Capitolo 6).



## 2. La modellazione fluidodinamica in ambito cromatografico

In questo capitolo viene presentato il contesto operativo all'interno del quale è stato svolto il presente lavoro, illustrando dapprima le applicazioni tecnologiche a cui questo si riferisce e in seguito gli strumenti utilizzati per eseguire l'analisi. In primo luogo, viene trattato il ruolo della cromatografia liquida nell'industria di processo, in particolare farmaceutica, come strumento per la separazione di soluzioni, miscele o sospensioni colloidali e le potenzialità modellistiche e applicative della fluidodinamica computazionale. In secondo luogo, si presentano alcuni degli aspetti teorici fondamentali propri dei fenomeni di trasporto e del calcolo numerico applicato al moto dei fluidi. Il capitolo si conclude esponendo i principali approcci con cui viene affrontato il problema della descrizione del moto di fluidi all'interno di mezzi porosi, quali in generale le colonne cromatografiche, riportando lo stato dell'arte della trattazione alla scala del poro, la tecnica più recente e dalle maggiori prospettive di sviluppo.

### 2.1 La cromatografia nell'industria farmaceutica

L'industria farmaceutica è un ambito molto vasto; essa comprende un numero grandissimo di prodotti, spesso costituiti da molecole organiche complesse e di grandi dimensioni. Poiché la sintesi per via chimica risulta frequentemente proibitiva o impraticabile, molti di questi prodotti, tra cui le preparazioni vaccinali, sono realizzati a partire da processi biotecnologici.

L'ambito dei vaccini è attualmente considerato particolarmente attrattivo per il mondo dell'industria, poiché la domanda globale è cresciuta in modo importante nei passati decenni ed è in crescita tuttora, sia per quanto riguarda l'uomo che per gli animali da allevamento. I progressi effettuati nella conoscenza della patogenesi di alcune importanti malattie infettive e nelle tecnologie manifatturiere hanno permesso di studiare e produrre vaccini efficaci nel contrasto di più di 25 microorganismi patogeni per l'essere umano [4]. Uno studio condotto nel 2003 stima che questi prevengano a livello globale poco meno di sei milioni di morti ogni anno [5], comportando inoltre importanti benefici sociali come la riduzione dei costi sanitari, l'aumento dell'aspettativa di vita e la promozione della crescita economica. La vaccinazione sfrutta i meccanismi naturali di difesa espletati attraverso il sistema immunitario, di conseguenza una preparazione vaccinale deve contenere componenti antigenici i quali possono essere il patogeno stesso, una sua parte o un derivato di esso. A questi sono spesso combinati dei coadiuvanti, aventi lo scopo di massimizzare la risposta immunitaria all'antigene bersaglio. Tradizionalmente, la vaccinazione è stata applicata per prevenire malattie veicolate da agenti quali virus, batteri e parassiti, ma essa appare uno strumento promettente anche nei confronti di condizioni patologiche autoimmuni o di origine neoplastica. Le preparazioni tradizionali sono ascrivibili alle seguenti categorie, suddivise in funzione del componente antigenico [6]:

- batteri vivi in forma attenuata;
- batteri morti o inattivati;
- virus attivi in forma attenuata;
- virus inattivati;
- anatossine (tossine inattivate);
- porzioni antigeniche del patogeno, generalmente presenti sulla superficie di quest'ultimo, come polisaccaridi, proteine e glicoproteine;
- più recentemente, precursori di componenti antigenici, in particolare porzioni di codice genetico contenenti le istruzioni per la sintesi a livello cellulare del componente antigenico bersaglio.

In tutti i casi presentati la componente antigenica deriva da un processo fermentativo, nel quale l'agente patogeno o un vettore ricombinante in grado di esprimere l'antigene bersaglio vengono prodotti in quantità tale da soddisfare le richieste di mercato in funzione dell'efficienza di estrazione dell'antigene nel conseguente processo di purificazione. I recenti progressi nell'espressione genica e nella tecnologia di coltura cellulare hanno fatto sì che la componente limitante del processo produttivo diventasse non più la fase fermentativa quanto piuttosto le fasi successive [7]. Oltre a ciò, attualmente i processi di *downstream* nella produzione di vaccini e vettori virali pesano per circa il 70% del costo dell'intero processo produttivo [8]. A titolo di esempio, si riporta in Fig. 2.1 lo schema del processo produttivo di un vaccino contro il virus dell'epatite B.

Un discorso analogo è applicabile alla produzione di anticorpi monoclonali a partire da ibridomi (cellule ibride realizzate attraverso la fusione di linfociti B e cellule neoplastiche); essi rappresentano una delle aree di ricerca farmacologica più promettenti, poiché sono in grado di generare una risposta immunitaria con altissima specificità verso un preciso *target* molecolare. Gli anticorpi sono infatti un particolare tipo di proteine, dotate di siti di legame specifici per un determinato antigene. La produzione a livello industriale di anticorpi, estratti da linfociti B di organismi precedentemente immunizzati nei confronti del patogeno bersaglio, permette di ottenere una terapia estremamente flessibile e applicabile ad una moltitudine di patologie differenti. Anche in questo caso, il processo produttivo contiene generalmente più di una separazione cromatografica, in particolare di affinità (basato sull'interazione specifica tra anticorpi e proteina A immobilizzata su particelle di resina), a scambio ionico e, meno frequentemente, ad interazione idrofobica. A titolo di esempio, si riporta in Fig. 2.2 lo schema del processo produttivo di un generico anticorpo monoclonale.

Generalmente, ciò che si ottiene in coda a un bioreattore è una miscela acquosa complessa costituita da cellule intere o frammentate e innumerevoli molecole organiche, proteine, lipidi e acidi nucleici. Gli elevati standard qualitativi richiesti in questo tipo di preparazioni, in termini di purezza, sterilità e assenza di contaminanti dannosi, rende necessaria una grande attenzione in fase di *downstream*. Il procedimento di estrazione e purificazione del prodotto di interesse si costituisce in una serie di passaggi aventi rispettivamente lo scopo di [9]:

- allontanare il particolato insolubile;
- effettuare una separazione primaria;
- purificare il prodotto di interesse;
- ottenere il prodotto finito in una forma stabile.

La sequenza di processi è differente nel caso di prodotti intracellulari o extracellulari, e viene ottimizzata per ogni singola molecola di interesse. Nonostante questo, la fase di purificazione prevede nella maggior parte dei casi almeno una separazione cromatografica [10], in particolar modo per le biomacromolecole; nell'ambito dei vaccini, il processo produttivo ne coinvolge mediamente tre [11]. Ciò perché, nonostante il costo possa essere significativo, permettono di ottenere alte selettività e una rimozione molto efficace delle impurezze [12]. La cromatografia rappresenta inoltre un ambito molto promettente nel perfezionamento dei processi produttivi biotecnologici, poiché i recenti sviluppi nell'ambito della tecnica di affinità potrebbero permettere l'elaborazione di procedure standardizzate di *downstream*, il cui passaggio chiave consiste in una separazione basata su ligandi ingegnerizzati a elevata affinità e selettività nei confronti del soluto bersaglio. In particolare, l'obiettivo può essere raggiunto attraverso procedure automatizzate di ricerca e selezione dei ligandi (si riporta un esempio della procedura in Fig. 2.3) e l'individuazione di strategie produttive di questi ad costo ridotto.

La cromatografia è una procedura di separazione atta a risolvere miscele, isolandone i componenti. Il principio che muove la separazione è la diversa velocità di migrazione dei componenti e il ritardo selettivo delle molecole di soluto durante il passaggio attraverso il mezzo cromatografico. All'interno dell'unità di processo distinguiamo due fasi: la fase stazionaria, ovvero sia il materiale solido che permane nella colonna e genera la separazione, e la fase mobile, ovvero sia la fase fluida che trasporta i soluti all'interno dell'apparecchiatura [14]. Per sua natura, la cromatografia è un processo discontinuo, di alta risoluzione e bassa potenzialità. Per questo motivo, all'interno dell'industria di processo trova le maggiori applicazioni negli ambiti della chimica fine, e in particolare farmaceutica. Le tecniche cromatografiche sono utilizzate nella separazione e purificazione di proteine, amminoacidi, acidi nucleici, alcaloidi, vitamine, steroidi e molte altre molecole biologiche, e possono essere suddivise in 7 macrocategorie [15]:

- di adsorbimento (ADC);
- di ripartizione liquido-liquido (LLC);
- a scambio ionico (IEC);
- a esclusione dimensionale (SEC);
- di affinità (AFC);
- a interazione idrofobica (HIC);
- ad alta pressione (HPLC).

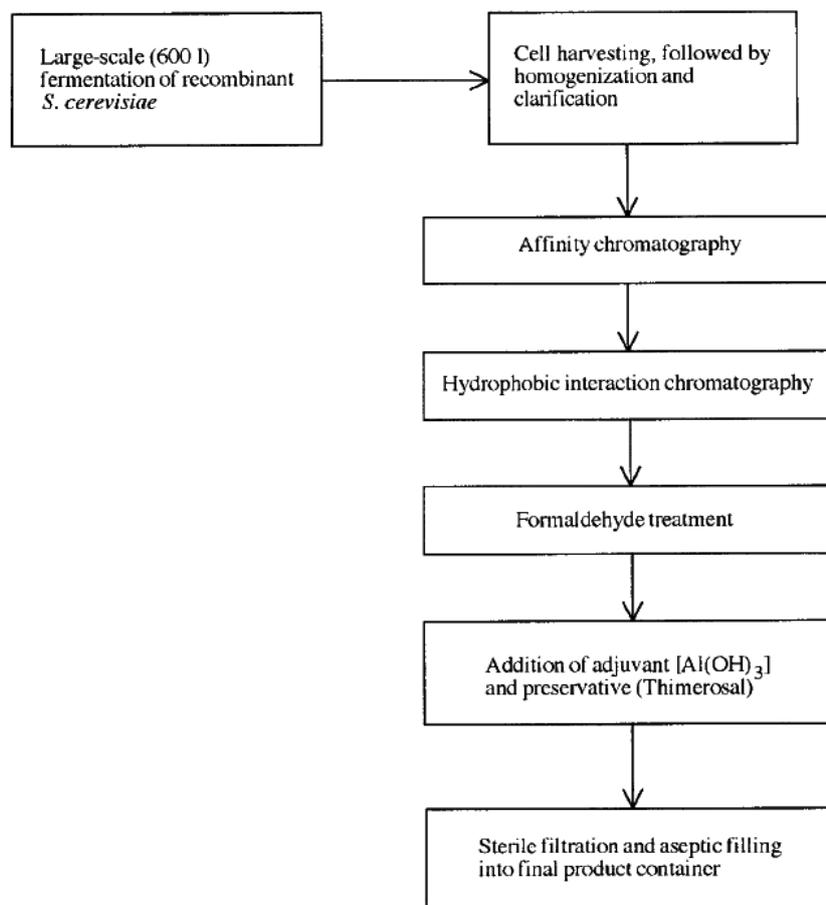


Figura 2.1: Schematizzazione del processo produttivo del vaccino ricombinante contro l'epatite B (Recombivax HB, Merck) [6].

Dal punto di vista tecnico, un'apparecchiatura cromatografica è costituita da una pompa, un campionatore, la colonna cromatografica vera e propria e un rivelatore, posti in serie. La colonna è generalmente un cilindro cavo, riempito in parte dalla fase stazionaria. Nel caso della separazione di proteine (HIC, IEC e SEC), la fase stazionaria è costituita generalmente da un impaccamento non strutturato di particelle sferiche, di materiale polimerico o ceramico, dotate di un'elevata porosità interna. In Fig. 2.4 si riporta una rappresentazione dei meccanismi di trasporto coinvolti in questo tipo di colonne. Esistono altre tipologie di fasi stazionarie, in particolare monolitiche e a membrana, tuttavia l'impaccamento casuale di sfere permette di ottenere maggiori risoluzioni e capacità di legame dinamico (DBC) superiori ed è il metodo più studiato e validato. Per questi motivi, esso trova una maggior diffusione in ambito industriale [12].

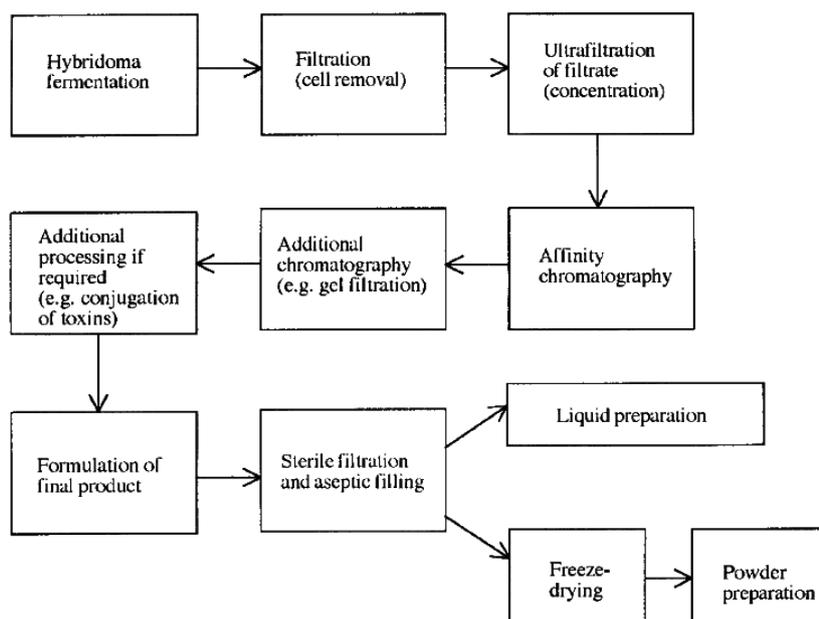


Figura 2.2: Schematizzazione del processo produttivo di un generico anticorpo monoclonale [6].

## 2.2 Nozioni fondamentali di fluidodinamica computazionale

La fluidodinamica computazionale, o CFD, è un insieme di metodi numerici funzionali alla risoluzione delle equazioni di bilancio di quantità di moto, energia e materia. Poiché le equazioni di bilancio sono equazioni differenziali alle derivate parziali (PDE) non lineari, la risoluzione numerica necessita la discretizzazione del dominio spaziale e temporale e di opportuni metodi di interpolazione delle grandezze all'interno del dominio [16].

### 2.2.1 Equazioni di governo

Di seguito vengono presentate le equazioni utilizzate nel corso del presente lavoro di tesi per descrivere il trasporto di quantità di moto e di energia. Per la risoluzione, sono state adottate le seguenti ipotesi:

- Dominio fluido:
  - fluido continuo, omogeneo, isotropo e incomprimibile;
  - fluido newtoniano;
  - flusso termico diffusivo espresso attraverso la legge di Fourier;
  - sono trascurati gli effetti dovuti alla gravità;

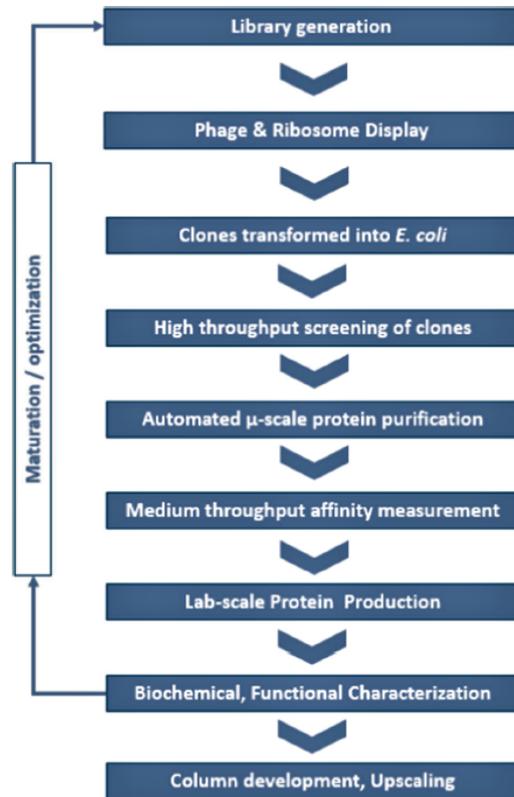


Figura 2.3: *Workflow* per la ricerca e l'individuazione di ligandi per la cromatografia di affinità [13].

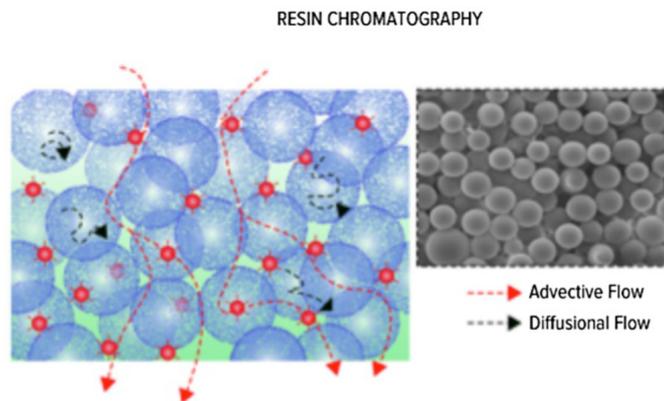


Figura 2.4: Rappresentazione schematica dei meccanismi di trasporto presenti all'interno di un riempimento non strutturato di particelle di resina cromatografica [13].

- densità, viscosità, conducibilità termica e calore specifico costanti nell'intervallo di temperatura considerato;
- è trascurato il termine energetico legato alla dissipazione viscosa nel fluido.
- Dominio solido:
  - solido pseudo-omogeneo, continuo, isotropo;
  - flusso termico diffusivo espresso attraverso la legge di Fourier;
  - densità, conducibilità termica efficace e calore specifico costanti nell'intervallo di temperatura considerato;
  - densità, calore specifico e conducibilità termica efficace approssimati attraverso l'interpolazione lineare delle proprietà termofisiche del fluido e della resina, pesate in funzione della porosità interna delle particelle.

Le equazioni risolte, qui presentate in forma vettoriale, sono:

- equazione di continuità:

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1)$$

dove  $\mathbf{u}$  è il vettore velocità;

- equazione di bilancio della quantità di moto, o equazione di Navier-Stokes:

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\rho(\nabla \cdot \mathbf{u}\mathbf{u}) - \nabla p - \mu \nabla^2 \cdot \mathbf{u}, \quad (2.2)$$

dove  $\rho$  è la densità del fluido,  $p$  la pressione e  $\mu$  è la viscosità dinamica, costante in accordo alla legge di Newton;

- equazione di bilancio dell'entalpia nel dominio fluido:

$$\rho \frac{\partial \hat{H}_f}{\partial t} = -\rho(\nabla \cdot \hat{H}_f \mathbf{u}) - k_f \nabla^2 T; \quad (2.3)$$

- Equazione di bilancio dell'entalpia nel dominio solido:

$$\rho \frac{\partial \hat{H}_s}{\partial t} = -k_s \nabla^2 T, \quad (2.4)$$

dove  $\hat{H}$  è l'entalpia specifica della fase considerata, e  $k$  la rispettiva conducibilità termica.

### 2.2.2 Metodi numerici

La risoluzione numerica di equazioni alle derivate parziali richiede la suddivisione del dominio di calcolo in un numero elevato di sottodomini geometricamente più semplici. Questo avviene attraverso la costruzione di una griglia tridimensionale, la quale va a descrivere una serie di piccoli volumi, detti celle. Ogni cella è racchiusa all'interno di facce condivise con le celle adiacenti, attraverso le quali interagisce con queste ultime.

Per ciò che riguarda il moto di fluidi, il metodo numerico maggiormente utilizzato per la risoluzione delle equazioni di bilancio è il metodo ai volumi finiti (FVM). Rispetto ad altri metodi di risoluzione (come il metodo agli elementi finiti o il metodo alle differenze finite), esso viene preferito per la sua proprietà di garantire la conservazione locale delle quantità trasportate. La discretizzazione del volume introduce necessariamente degli errori numerici, la cui entità è proporzionale alla dimensione delle celle. Poiché però la diminuzione della dimensione di cella rende necessario generarne un numero maggiore, la riduzione dell'errore avviene con un aggravio del costo computazionale. La condizione di ottimo è propria di ogni singolo caso in esame, e deve considerare il fatto che l'errore di simulazione non può venire ridotto oltre un certo limite, dovuto all'aritmetica dell'elaboratore (il cosiddetto errore di troncamento). Il principio su cui si basa il metodo ai volumi finiti consiste nell'esprimere le equazioni di bilancio, precedentemente presentate in forma differenziale, in forma integrale, approssimarle attraverso equazioni algebriche, e infine applicarle ad ogni singolo sottodominio. Di seguito viene riportato il processo di elaborazione dell'equazione di bilancio di una generica quantità  $\phi$ , dove  $\rho$  è la densità, assunta costante, mentre  $\Gamma$  è il coefficiente di trasporto. La trattazione segue il testo [16]. L'equazione di trasporto assume la forma:

$$\rho \frac{\partial \phi}{\partial t} = -\rho \nabla \cdot (\mathbf{u}\phi) + \nabla \cdot (\Gamma \nabla \phi) + S_\phi. \quad (2.5)$$

Integrando l'Eq. (2.5) sul volume di controllo  $\omega$  si ottiene:

$$\int_\omega \rho \frac{\partial \phi}{\partial t} dV = - \int_\omega \rho \nabla \cdot (\mathbf{u}\phi) dV + \int_\omega \nabla \cdot (\Gamma \nabla \phi) dV + \int_\omega S_\phi dV. \quad (2.6)$$

All'Eq. (2.6) si applica il teorema di Gauss. Nell'equazione che segue,  $S$  è la superficie chiusa che contiene  $\omega$ .

$$\int_\omega \rho \frac{\partial \phi}{\partial t} dV = - \int_S \rho \mathbf{u} \cdot \mathbf{n} \phi dS + \int_S \Gamma \nabla \phi \cdot \mathbf{n} dS + \int_\omega S_\phi dV. \quad (2.7)$$

Per poter riformulare l'Eq. (2.7) in termini algebrici, devono essere introdotte alcune semplificazioni. Il valore della proprietà viene calcolato esclusivamente nel centroide di ogni cella e viene considerato uniforme all'interno di questa. Il valore del flusso di  $\phi$  viene calcolato esclusivamente al centro di ogni faccia, e viene considerato uniforme sull'intera superficie della faccia. Alla luce delle semplificazioni di cui sopra, il termine di accumulo diviene:

$$\int_\omega \rho \frac{\partial \phi}{\partial t} dV \approx \rho \frac{\partial \phi}{\partial t} V. \quad (2.8)$$

Il termine convettivo diviene:

$$\int_S \rho \mathbf{u} \cdot \mathbf{n} \phi dS \approx -\rho \sum_{i=1}^N (A \mathbf{u} \cdot \mathbf{n} \phi)_i, \quad (2.9)$$

in cui  $N$  rappresenta il numero di facce della cella,  $A$  l'area della faccia,  $\mathbf{n}$  è il versore normale alla faccia e  $\phi$  è considerato uniforme sulla faccia. Il termine diffusivo viene trattato in maniera analoga a quello convettivo:

$$\int_S \Gamma \nabla \phi \cdot \mathbf{n} dS \approx - \sum_{i=1}^N (A \Gamma \nabla \phi \cdot \mathbf{n})_i. \quad (2.10)$$

Poiché contiene un termine differenziale, si rende necessaria un'ulteriore semplificazione attraverso lo sviluppo di Taylor di primo ordine del gradiente. Il termine sorgente diviene:

$$\int_{\omega} S_{\phi} dV \approx \bar{S}_{\phi} V. \quad (2.11)$$

Nel complesso, si ottiene:

$$\rho \frac{\partial \phi}{\partial t} V = \rho \sum_{i=1}^N (A \mathbf{u} \cdot \mathbf{n} \phi)_i - \sum_{i=1}^N (A \Gamma \nabla \phi \mathbf{n})_i + \bar{S}_{\phi} V. \quad (2.12)$$

I valori di  $\phi$  e del suo gradiente in corrispondenza delle facce sono valutati attraverso una funzione detta schema di interpolazione. I codici di fluidodinamica computazionale generalmente rendono disponibili diversi schemi di interpolazione, che vanno scelti in funzione delle caratteristiche del sistema fisico in esame come anche della disponibilità di risorse di calcolo. Considerando un sistema composto da due celle adiacenti, aventi una faccia in comune e allineate alla direzione del moto, attraverso le quali si ha il trasporto di una proprietà  $\phi$ , si possono identificare due casi estremi:

- una situazione puramente convettiva, dove la proprietà è trasportata esclusivamente dal moto del fluido;
- una situazione puramente diffusiva, dove il flusso di  $\phi$  è linearmente proporzionale al suo gradiente.

Nel primo caso, appare naturale identificare il valore di  $\phi$  sulla faccia con il valore calcolato nel centroide della cella a monte del moto, mentre nel secondo è opportuno esprimere il valore sulla faccia attraverso l'interpolazione lineare di  $\phi$  tra i due centroidi delle celle. Tutti gli schemi di interpolazione disponibili si inseriscono all'interno dell'intervallo identificato dai due casi presentati sopra, e vanno scelti attraverso considerazioni su quale sia il meccanismo di trasporto dominante all'interno del sistema. Nelle simulazioni effettuate sono stati utilizzati principalmente due schemi:

- *Central Differencing Scheme* (CDS) per valutare i gradienti e il laplaciano;
- *Linear Upwind Differencing Scheme* (LUDS), in particolare la sua formulazione *bounded*, per valutare le divergenze.

Lo schema CDS consiste nell'interpolazione lineare della quantità  $\phi$  o del suo gradiente. Viene presentato qui in forma semplificata, considerando un caso unidimensionale, lungo l'asse  $x$ , e seguendo la notazione di Fig. 2.5. Sulla faccia  $w$ , i valori di  $\phi$  e del suo gradiente vengono calcolati secondo le seguenti equazioni:

$$\phi_w = \frac{x_w - x_P}{x_W - x_P} \phi_P + \frac{x_W - x_w}{x_W - x_P} \phi_W, \quad (2.13)$$

$$\left( \frac{d\phi}{dx} \right)_w = \frac{\phi_W - \phi_P}{x_W - x_P}. \quad (2.14)$$

Lo schema LUDS viene utilizzato per valutare il valore di  $\phi$  sulla faccia tenendo in considerazione l'effetto del trasporto per convezione. Per il calcolo vengono utilizzati i valori di  $\phi$  delle due celle "a monte" della faccia, ovvero le due celle incontrate muovendosi in direzione opposta al vettore  $\mathbf{u}$  partendo dalla faccia. Lo schema assume che:

$$\left(\frac{d\phi}{dx}\right)_e = \left(\frac{d\phi}{dx}\right)_w ; \quad (2.15)$$

di conseguenza, il valore di  $\phi$  sulla faccia  $w$  viene calcolato secondo l'equazione:

$$\phi_w = \phi_P + \frac{(\phi_P - \phi_E)(x_w - x_P)}{x_P - x_E} . \quad (2.16)$$

Lo schema LUDS non è limitato, ovvero non è in grado di garantire che il valore calcolato sulla faccia sia compreso tra i valori nei centri delle celle circostanti. Per favorire la convergenza della soluzione, nel corso del presente lavoro è stata utilizzata la variante "bounded" dello schema LUDS, utilizzabile nel caso di fluidi incomprimibili. La variante prevede l'inclusione della divergenza della velocità nel termine convettivo. Quest'ultima è nulla per un fluido incomprimibile, ma può assumere valori diversi da zero nel corso del processo iterativo di soluzione, tendendo a zero al raggiungimento della convergenza [17]. In termini matematici, il termine convettivo viene calcolato applicando la condizione in Eq. (2.17) all'Eq. (2.18).

$$\nabla \cdot \mathbf{u} \neq 0 , \quad (2.17)$$

$$\rho \mathbf{u} \cdot \nabla \phi = \rho \nabla \cdot (\mathbf{u} \phi) - \rho (\nabla \cdot \mathbf{u}) \phi . \quad (2.18)$$

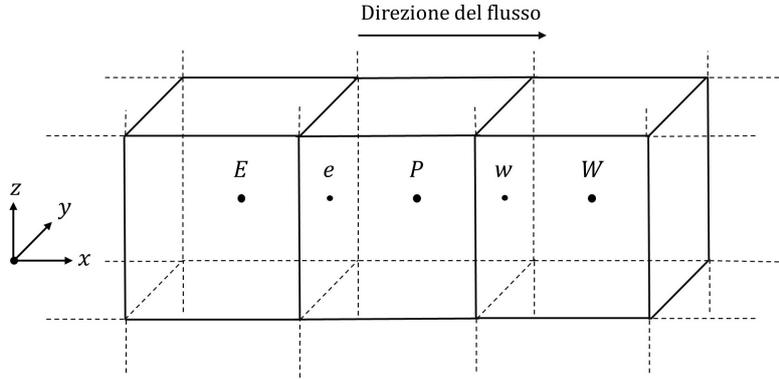


Figura 2.5: Esempio di celle tridimensionali esaedriche.

Elaborando in questo modo il sistema fisico in esame e le equazioni di bilancio, il problema si riduce a un sistema di equazioni algebriche, risolubile per via iterativa. Nell'equazione del moto è presente un termine sorgente relativo al gradiente di pressione. Poiché nelle equazioni di bilancio non è presente un'equazione esplicita per il campo di pressione, a

causa dell'assunzione di fluido incompressibile, si rende necessario utilizzare un algoritmo aggiuntivo che vada a risolverlo iterativamente. Gli algoritmi di *pressure-velocity coupling* risolvono l'equazione di Poisson, riportata in Eq. (2.19), derivata dall'equazione di Navier-Stokes applicando l'operatore divergenza e sostituendo l'equazione di continuità, ottenendo un nuovo campo di pressione dato il campo di velocità:

$$\frac{\partial}{\partial x_i} \left( \frac{\partial p}{\partial x_i} \right) = - \frac{\partial}{\partial x_i} \left( \frac{\partial(\rho u_i u_j)}{\partial x_j} \right). \quad (2.19)$$

Nel corso del presente lavoro di Tesi si è impiegato per l'accoppiamento tra campo di moto e pressione l'algoritmo SIMPLE. Esso utilizza un'ipotesi di partenza del campo di pressione, mediante la quale viene risolta l'equazione del moto. Successivamente, il campo di pressione viene calcolato a partire dal campo di moto attraverso l'equazione di Poisson. Il processo prosegue per via iterativa, aggiornando il campo di moto e di pressione, fino al raggiungimento della convergenza.

La dipendenza temporale dell'equazione di trasporto viene risolta suddividendo l'intervallo temporale simulato in un certo numero di intervalli discreti, di ampiezza  $\Delta t$ , e successivamente integrando l'equazione sull'intervallo, considerando le semplificazioni precedentemente adottate. Adottando questo principio all'Eq. (2.12), si ottiene la seguente equazione:

$$\int_t^{t+\Delta t} \rho \frac{\partial \phi}{\partial t} V dt = \int_t^{t+\Delta t} \rho \sum_{i=1}^N (\mathbf{A}\mathbf{u} \cdot \mathbf{n}\phi)_i dt - \int_t^{t+\Delta t} \rho \sum_{i=1}^N (\mathbf{A}\mathbf{u} \cdot \mathbf{n}\phi)_i dt + \int_t^{t+\Delta t} \bar{S}_\phi V dt, \quad (2.20)$$

che diventa:

$$\rho V [\phi|_{t+\Delta t} - \phi|_t] = \int_t^{t+\Delta t} \left( \rho \sum_{i=1}^N (\mathbf{A}\mathbf{u} \cdot \mathbf{n}\phi)_i - \rho \sum_{i=1}^N (\mathbf{A}\mathbf{u} \cdot \mathbf{n}\phi)_i + \bar{S}_\phi \right) dt. \quad (2.21)$$

I metodi numerici per la risoluzione di problemi transitori possono essere raggruppati in tre categorie: espliciti, impliciti e ibridi. Essi differiscono nel calcolo della derivata temporale al tempo  $t$ . In generale, l'integrale di una quantità  $\phi$  nel tempo può essere approssimato secondo l'equazione:

$$\int_t^{t+\Delta t} \phi dt \approx [\theta \phi|_{t+\Delta t} + (1 - \theta) \phi|_t] \Delta t, \quad (2.22)$$

dove  $\theta$  è un fattore di peso. Applicando questo ragionamento all'Eq. (2.21), si distinguono tre differenti casi in funzione del valore di  $\theta$ . Per  $\theta = 0$ , il termine a sinistra viene calcolato utilizzando esclusivamente  $\phi|_t$ , in questo modo si ottiene la cosiddetta discretizzazione esplicita. Viceversa, per  $\theta = 1$  viene considerato esclusivamente il nuovo valore di  $\phi$ , ottenendo la discretizzazione implicita. Per tutti gli altri valori di  $\theta$ , ci si trova in una situazione intermedia. I metodi impliciti godono rispetto agli altri di un vantaggio sostanziale; essi necessitano della risoluzione per via iterativa (a differenza dei metodi espliciti) ma sono incondizionatamente stabili. Nel corso del presente lavoro si è utilizzato il metodo di Eulero implicito, caratterizzato da errore di discretizzazione di

primo ordine in  $\Delta t$ . Per quanto il metodo scelto sia incondizionatamente stabile, è buona norma scegliere un intervallo temporale minore del tempo caratteristico di trasporto di  $\phi$  attraverso la cella, e di conseguenza rispettare la condizione data dalle equazioni seguenti, applicate rispettivamente al dominio fluido e al dominio solido:

$$\text{Co} = \frac{\Delta x}{u\Delta t} < 1, \quad (2.23)$$

$$N_{\text{diff}} = \frac{\rho(\Delta x)^2}{\Gamma\Delta t} < 1, \quad (2.24)$$

in cui  $\Delta x$  rappresenta la dimensione caratteristica di cella. Questa condizione è detta di Courant-Friedrichs-Lewy o CFL. Nel corso del presente lavoro i *timesteps* sono stati scelti in modo che il numero di Courant medio fosse minore di uno; non è stato possibile rispettare questa condizione anche per il numero di diffusione, poiché avrebbe comportato un aggravio computazionale eccessivo.

### 2.2.3 Griglie di calcolo

Le griglie di calcolo vengono suddivise in tre categorie: strutturate, non strutturate e ibride. La prima si riferisce a griglie, cartesiane o *body fitted*, costituite da elementi quadrangolari in 2D ed esaedrici in 3D. In una griglia strutturata i vertici sono collocati all'intersezione delle linee della griglia e possono essere espressi attraverso una matrice, inoltre i punti interni hanno un numero costante di vertici circostanti. In generale, l'adozione di una griglia di questo tipo è preferibile, poiché consente una riduzione delle risorse di calcolo e di memoria impiegate in confronto ad altre tipologie. Tuttavia, essa è poco flessibile e nel caso di geometrie complesse la costruzione di una griglia strutturata può risultare impossibile. Una griglia non strutturata è formata da elementi poliedrici di forma arbitraria, arbitrariamente disposti nello spazio. In questo modo è possibile descrivere qualsiasi tipo di geometria, anche le più complesse, con un grado di raffinamento predefinito e variabile all'interno della geometria simulata. Le *mesh* ibride combinano entrambi gli approcci, essendo composte da regioni strutturate e regioni non strutturate. Nel corso del presente lavoro di tesi le griglie di calcolo sono state costruite utilizzando l'*utility* `snappyHexMesh`, contenuta nel pacchetto `OpenFOAM`, di cui si espone brevemente il funzionamento [18]. `snappyHexMesh` genera automaticamente *mesh* tridimensionali costituite da elementi esaedrici e *split-hexahedra*, ovverosia frazioni di celle esaedriche, a partire da geometrie generiche espresse come superfici costituite da triangoli (files STL o OBJ). La griglia viene costruita tramite un processo iterativo, a partire da una *background mesh* strutturata. Il procedimento di costruzione, illustrato in Fig. 2.6, prevede:

- La selezione delle celle intersecate dalla superficie del solido tridimensionale e/o all'interno di una regione specificata dall'utente;
- la divisione (lungo gli assi) delle celle selezionate in sottocelle, successivamente selezionate e suddivise a loro volta fino al raggiungimento del livello di raffinamento richiesto dall'utente;
- l'eventuale rimozione delle celle non appartenenti al dominio di simulazione;

- l'adattamento delle celle adiacenti alla superficie attraverso una modifica della loro forma.

Il vantaggio di questo metodo è la possibilità di creare griglie costituite prevalentemente da elementi esaedrici, strutturati, pur essendo in grado di descrivere geometrie anche complesse.

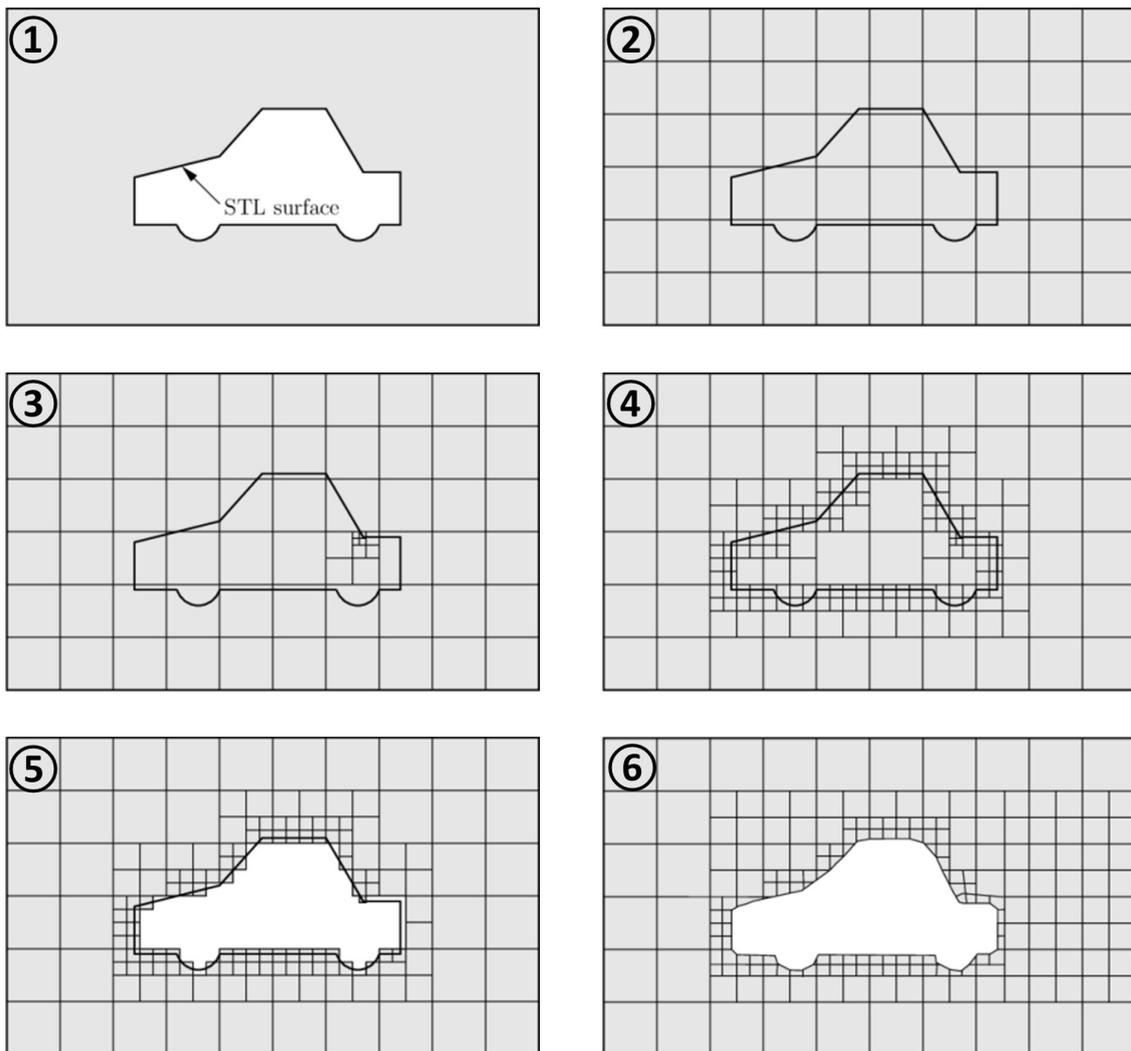


Figura 2.6: Schematizzazione della procedura di *meshing* della utility `snappyHexMesh` [18].

## 2.3 Modellazione di colonne cromatografiche

In ambiti industriali, vengono universalmente utilizzati due approcci differenti nello sviluppo di processi di separazione:

- utilizzo diretto dei dati sperimentali;
- costruzione di modelli meccanicistici più o meno articolati, spesso frutto della combinazione di principi primi e considerazioni empiriche.

Il primo approccio non richiede una conoscenza dettagliata del fenomeno ed è in grado di catturare i parametri critici che determinano la qualità del prodotto ottenuto; richiede tuttavia la raccolta di una larga mole di dati empirici e ciò si traduce in un consistente impiego di tempo e materiali. La modellazione quantitativa della cromatografia, applicata all'ottimizzazione delle condizioni operative, può contribuire significativamente alla riduzione dei costi di sviluppo del processo di separazione e a migliorarne le prestazioni [19]. La conoscenza dei processi fornita da modelli meccanicistici contribuisce a creare uno spazio di condizioni operative all'interno del quale potersi muovere alla ricerca dei parametri ottimali, in un'ottica di ottenimento della *quality by design* (QbD) [20]. Qualitativamente, i fenomeni di trasporto e i fenomeni di adsorbimento sono in larga parte indipendenti, ciò giustifica il loro studio separato. In ogni caso, in entrambi i casi i modelli matematici generalmente impiegati trovano fondamento in importanti semplificazioni fisiche e una forte base empirica, sia perché le interazioni tra i soluti e la fase stazionaria sono specifiche per ogni coppia soluto/fase stazionaria, sia per le caratteristiche peculiari del moto all'interno dell'impaccamento. Un modello completo deve necessariamente adottare un approccio multiscala, in cui le prestazioni dell'intera colonna sono legate ai fenomeni che avvengono alla scala degli elementi che costituiscono la fase stazionaria, fino ad arrivare a livello molecolare. Per ciò che concerne i fenomeni di trasporto, la trattazione è comune a tutti i tipi di letto impaccato, e si possono identificare due strutture geometriche chiave per la loro analisi:

- lo spazio vuoto tra sfera e sfera di resina (porosità esterna);
- la porosità propria delle particelle di resina (porosità interna).

Considerata l'elevata complessità della struttura dei pori interni alle particelle, queste ultime vengono frequentemente considerate quali mezzi solidi, all'interno del quale il trasporto avviene per diffusione (*effective medium models* [21]), oppure quali mezzi fluidi pseudo-omogenei, descrivendo il flusso al loro interno attraverso equazioni quali l'equazione di Brinkman [22]. La descrizione della fluidodinamica a livello della porosità esterna del letto ha ricevuto un crescente interesse negli scorsi decenni in virtù del suo ruolo determinante nella cinetica del trasporto di materia, che insieme a cinetica e termodinamica di adsorbimento/desorbimento dei soluti, è ciò che determina la capacità di separazione del letto. In ambiti industriali, l'approccio adottato con maggior frequenza per lo sviluppo di un sistema di separazione cromatografica è l'*up-scaling* a partire da colonne di laboratorio o impianti pilota, ottenuto generalmente aumentando il diametro della colonna a parità di altezza del letto e di velocità spaziale del fluido. La difficoltà legata a questo tipo di approccio risiede nell'assicurare il mantenimento della stessa qualità nella distribuzione del flusso passando da una scala alla superiore [10]. Per questo motivo, a partire dalla fine degli anni '90 sono stati pubblicati studi nella quale si indagavano diversi aspetti legati al flusso e alla sua distribuzione all'interno della colonna attraverso la fluidodinamica computazionale [11]. L'ottenimento dei parametri di interesse attraverso modelli

meccanicistici permette di ottenere risultati virtualmente indipendenti dalle condizioni di distribuzione del flusso proprie della colonna utilizzata, risultando in una maggiore applicabilità.

### 2.3.1 Modellazione alla scala della colonna

Le simulazioni convenzionali di flusso in un mezzo poroso sono basate sul concetto di media spaziale e temporale delle quantità di interesse, ovverosia definendo le velocità e le direzioni del trasporto di queste quantità in un volume discreto (detto *Representative Elementary Volume*, REV) all'interno del quale gli effetti della struttura geometrica effettiva sulle variazioni del flusso e del trasporto di materia e calore vengono approssimati con modelli di chiusura, risultato di un'operazione di media sul volume [23]. In questo senso, non sono modelli ai principi primi ma necessitano l'ottenimento per via empirica dei parametri. Nel caso dei mezzi porosi, le correlazioni macroscale più usate sono la legge di Darcy e le sue successive implementazioni (come la legge di Darcy-Forchheimer o l'equazione di Brinkman) per descrivere il trasporto di quantità di moto e l'aggiunta di termini dispersivi nelle equazioni di bilancio di calore e materia. In questo modo, si ottiene un campo di velocità mediato, generalmente costante lungo l'asse ed eventualmente variabile lungo il raggio [24], attraverso il quale viene calcolato il trasporto di calore e materia attraverso le strategie di cui sopra. Il vantaggio sostanziale di una modellazione di questo tipo è il ridotto costo computazionale che questa comporta, tuttavia richiede la conoscenza a priori dei parametri mediati attraverso i quali descrivere il trasporto.

### 2.3.2 Modellazione alla scala delle particelle

La necessità di modelli che descrivano i fenomeni di trasporto alla scala delle particelle deriva dal fatto che il comportamento globale del sistema discende direttamente dalle condizioni locali del moto [25]. La disposizione casuale delle sfere all'interno della colonna genera una certa incertezza su alcuni parametri geometrici, quale ad esempio la distribuzione dimensionale dei canali tra una particella e l'altra, che influenzano in modo marcato la fluidodinamica del sistema e quindi anche il trasporto di calore e materia all'interno di esso. Di conseguenza, la modellazione "convenzionale" di una colonna richiede un elevato numero di esperimenti per ricavare informazioni come la perdita di carico associata ad un determinato flusso, la porosità esterna e interna, il coefficiente di ripartizione. Le procedure di creazione dell'impaccamento inoltre difficilmente sono riproducibili e in grado di garantire una adeguata uniformità di distribuzione delle particelle. La creazione di modelli basati invece su una descrizione dettagliata della geometria di una porzione rappresentativa del letto e del trasporto di quantità di moto, materia e calore all'interno di essa permettono di ottenere predizioni accurate sia sul grado di vuoto della colonna sia su parametri operativi senza richiedere prove empiriche al di fuori di quelle necessarie all'ottenimento dei parametri specifici relativi all'accoppiamento resina-soluto. La modellazione alla scala della particella può dunque essere un mezzo per ottenere informazioni da utilizzare nelle simulazioni dell'intera colonna [26], è stato infatti dimostrato come la modellazione CFD a livello delle particelle sia in grado di quantificare le componenti della dispersione idrodinamica nei letti impaccati su diverse scale spaziali e temporali [27]. Le

sfide maggiori per quanto riguarda la costruzione di modelli alla scala della particella riguardano la ricostruzione fedele della geometria del letto e la rappresentazione della risposta dinamica del letto in condizioni di flusso, nonché la trattazione dei fenomeni che avvengono nei pressi della parete della colonna, legati in particolar modo alla diminuzione della porosità media del letto in quella zona. Per quanto riguarda la costruzione della geometria, un approccio ben consolidato consiste nell'utilizzo di codici di calcolo basati sul metodo agli elementi discreti (DEM), che valuta traiettorie e velocità delle particelle mentre queste cadono per gravità e riempiono il volume preposto alla ricostruzione del letto. Il metodo prevede il calcolo delle forze agenti sulle singole particelle, dovute a gravità e impatti, lungo una suddivisione temporale discreta [21]. Un approccio alternativo consiste nell'utilizzo di metodi basati sulla dinamica di corpo rigido (CD, *Contact Dynamics* [28]), che applica le leggi della dinamica newtoniana considerando le entità geometriche come corpi rigidi, indeformabili [29, 30]. La semplificazione introdotta con l'ipotesi di corpo rigido trova grande utilità nella creazione di impaccamenti di elementi rigidi di forma arbitraria, tuttavia è possibile applicarla anche nella descrizione di letti cromatografici poiché alcuni studi evidenziano come la compattazione del letto cromatografico sia dovuta in larga parte al riarrangiamento delle particelle di resina, piuttosto che dalla loro deformazione [31], rendendo possibile trascurare gli effetti della deformabilità delle particelle di resina sulla geometria del letto.

### 2.3.3 Modellazione CFD dello scambio termico all'interno di letti impaccati: stato dell'arte

Il problema dello scambio termico all'interno di letti impaccati trova la sua principale applicazione nella descrizione di reattori catalitici. Il primo studio riguardante la modellazione alla scala della particella del trasporto di calore è stato effettuato su geometrie 2D da Dalman et al. [32], nel quale si è indagato il trasporto di calore in un tubo cilindrico contenente due sfere non a contatto (in 2D, un canale rettangolare contenente due circonferenze), valutando gli effetti della variazione dei numeri di Reynolds e Prandtl, del diametro delle sfere e della distanza tra esse, risolvendo direttamente l'equazione di Navier-Stokes (DNS, *Direct Numerical Simulation*). Lloyd e Bohem [33] hanno calcolato i numeri di Nusselt locali su una serie di 8 sfere immerse in un flusso di aria e di acqua, valutando l'effetto della distanza tra le sfere, anch'essi performando una simulazione assialsimmetrica 2D in DNS. Il primo studio in tre dimensioni del problema dello scambio termico in un letto impaccato è stato condotto da Derkx e Dixon [34], in cui si è calcolato il numero di Nusselt alla parete in diverse sezioni lungo l'asse di un canale cilindrico contenente tre sfere non a contatto, utilizzando il modello  $k - \epsilon$  per descrivere la turbolenza. Studi più recenti [35, 36] si sono occupati del trattamento di letti di sfere adiacenti, affrontando il problema del trattamento dei punti di contatto tra le sfere. Dal punto di vista geometrico infatti, considerando le particelle come sfere rigide il contatto tra esse avviene in uno spazio puntiforme. Nel caso si utilizzi un metodo ai volumi finiti per la risoluzione delle equazioni di trasporto, questo causa problemi in fase di creazione della griglia di calcolo, poiché nei pressi del punto di contatto le celle dovrebbero essere molto piccole o di forma estremamente allungata, implicando rispettivamente un aggravio

del costo computazionale o una diminuzione di stabilità del *solver*. Gli autori propongono diversi metodi di modifica dei punti di contatto, sia di tipo “globale” che di tipo “locale”, schematizzati in Fig 2.7. I metodi *Gaps* e *overlaps* consistono rispettivamente nel ridurre o aumentare il diametro effettivo delle particelle in modo tale che queste non risultino in contatto oppure che la superficie di contatto abbia area finita. Sono definiti metodi globali poiché hanno effetto sull’intera superficie della particella. Il metodo *bridges* prevede la sostituzione del punto di contatto con un solido cilindrico perpendicolare alla superficie delle sfere e avente diametro pari ad una frazione del diametro di queste ultime. Il metodo *caps* prevede la rimozione, in corrispondenza del punto di contatto, di una calotta sferica in entrambe le particelle, creando così un dominio fluido di ampiezza definita tra una sfera e l’altra. Come si evince dagli studi, i metodi di modifica locale

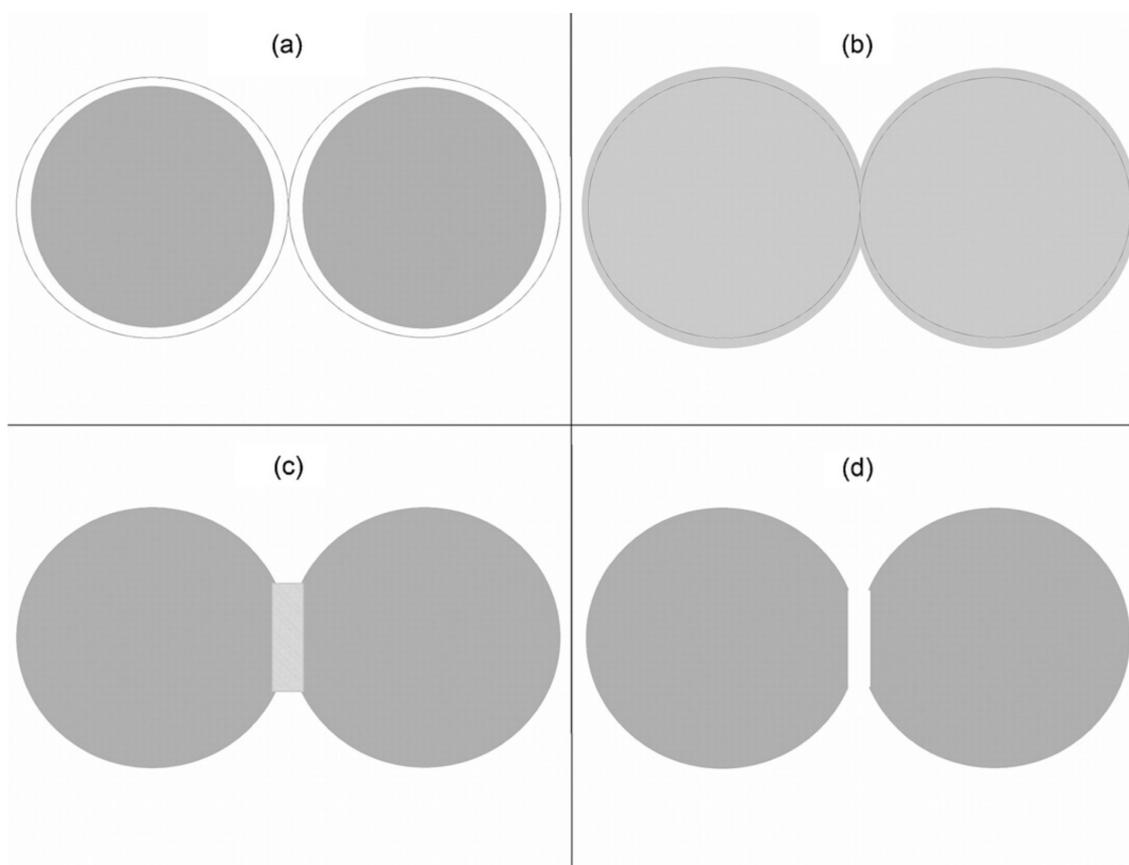


Figura 2.7: Schematizzazione dei metodi di modifica globale (casi a, b) e locale (casi c, d) dei punti di contatto: (a) *Gaps*; (b) *Overlaps*; (c) *Bridges*; (d) *Caps* [35].

risultano più idonei poiché, comportando la modifica di una frazione molto piccola del volume di simulazione, hanno un effetto molto limitato sul grado di vuoto del letto. Questo permette previsioni più accurate del campo di moto, poiché si è osservato che ad una modifica della porosità di un punto percentuale, mantenendo invariata la posizione dei centri delle sfere, corrisponde una variazione di circa il 3% sulla predizione delle perdite di

carico attraverso il letto. Lo studio dimostra inoltre come, per quanto riguarda i problemi di scambio termico, il metodo *bridges* fornisca risultati più accurati, in quanto trascurare la diffusione solido-solido porta a sottostimare il trasporto del calore all'interno del letto. Per evitare errori sulla predizione del campo di moto si suggerisce di imporre il raggio del cilindro interposto tra le particelle in modo tale che  $r_b < 0.1D_p$ ; inoltre la conducibilità termica dei cilindri è stata calcolata attraverso la relazione:

$$k_{eff} = \frac{2l_b}{r_b^2} \int_0^{r_b} \frac{k_s k_f}{h_s k_f + h_f k_s} r dr, \quad (2.25)$$

dove  $k_{eff}$  è la conducibilità termica efficace del *bridge*,  $l_b, r_b$ , sono la sua lunghezza e il suo raggio,  $k_s, k_f$  sono le conducibilità rispettivamente del solido e del fluido,  $h_s, h_f$  è la lunghezza del cammino di conduzione reale (ovverosia senza l'introduzione del *bridge*) nel solido e nel fluido, funzioni della posizione radiale. Si vuole mettere in evidenza che, nel caso a cui si riferisce lo studio, la differenza tra le conducibilità del fluido e del solido è di due ordini di grandezza. Nel caso presente, la conducibilità termica del fluido e delle sfere è molto simile, risultando in una conducibilità efficace dei *bridges* che differisce dalla conducibilità del solido meno del 10% di quest'ultima. Per questo motivo nel presente studio questo aspetto è stato trascurato, e ai *bridges* è stato assegnato lo stesso valore di conducibilità termica delle sfere.

### 3. I modelli *data-driven* e le loro applicazioni nell'ambito dell'industria di processo

Il concetto di intelligenza artificiale (AI) racchiude in sé un insieme molto vasto di concetti teorici e applicativi aventi come obiettivo comune lo sviluppo di sistemi in grado di imitare il processo cognitivo e decisionale degli esseri umani. Considerata la portata e la vastità del compito che si prefigge, la materia coinvolge e implementa conoscenze provenienti da diversi ambiti scientifici, dalle neuroscienze alla statistica. È un campo in continuo sviluppo, trovando applicazioni in qualsiasi settore della scienza e della tecnica, e viene declinato in numerosi metodi diversi. Il *machine learning* (ML) è un ramo dell'intelligenza artificiale che si occupa dello sviluppo di algoritmi in grado di costruire in modo autonomo relazioni tra i dati che gli vengono sottoposti, adattando e correggendo automaticamente le proprie caratteristiche, allo scopo di identificare all'interno di questi dati schemi ricorrenti ed effettuare classificazioni o previsioni. Non solo, alcune applicazioni sono in grado di utilizzare le relazioni tra i dati da loro stesse individuate per prendere delle decisioni anche in condizioni di incertezza [37]. I modelli costruiti attraverso tecniche di *machine learning* sono detti *data-driven*, poiché non richiedono la conoscenza a priori dei fenomeni fisici che regolano il sistema in analisi e delle relazioni causa-effetto tra le grandezze in gioco. Il *machine learning* sta incontrando un'attenzione crescente in moltissimi, se non in tutti, ambiti industriali, in virtù della sua grande flessibilità di applicazione. Considerata la crescente mole di dati a disposizione dell'industria, questo tipo di modelli soddisfa la necessità di un trattamento automatico di questi dati, in grado di raggrupparli, valutarne la significatività, scartare gli *outliers* e molte altre operazioni, in modo da estrarre informazioni utili per guidare i processi decisionali. Poiché l'industria di processo è spesso chiamata ad affrontare problemi complessi e determinati da una moltitudine di variabili, ha un forte interesse verso le potenzialità che questa tipologia di metodi offre, e li applica con successo in diversi campi, quali ad esempio: l'ottimizzazione dell'efficienza energetica [38], la costruzione di *digital twins* di apparecchiature e impianti [39], lo *smart manufacturing* [40]. È possibile individuare tre categorie di ML: *supervised learning*, *unsupervised learning* e *reinforcement learning*. Il primo è un approccio di tipo predittivo, in cui l'obiettivo è la definizione di una funzione, che dato un vettore  $\mathbf{x}$  in ingresso restituisce un vettore  $\hat{\mathbf{y}}$  in uscita. La funzione viene costruita a partire da un insieme noto di coppie input-output  $\mathcal{D} = (\mathbf{x}_j, \mathbf{y}_j)_{j=1}^N$ , dove  $\mathcal{D}$  è detto *training dataset*, mentre  $N$  è il numero di coppie che costituiscono l'insieme, dette *samples*. Gli elementi del vettore in ingresso sono detti *features*, essi sono i parametri che il costruttore del modello seleziona come significativi per la predizione dei parametri in uscita, questi ultimi detti *labels*. *Features* e *labels* possono essere sia variabili categoriche che scalari. È quindi possibile affrontare sia problemi di regressione che di classificazione, in questi ultimi si rende però necessario esprimere le variabili categoriche attraverso degli scalari. L'*unsupervised learning* è invece un approccio di tipo descrittivo, il cui scopo è identificare *pattern* ricorrenti e non prevedibili altrimenti.

In questo caso i dati in ingresso non possiedono *labels*, questa tecnica può essere applicata per raggrupparli sulla base di caratteristiche comuni (*clustering*), ridurre la dimensionalità dei *samples* o individuare eventuali *outliers* presenti nel dataset. Infine, la tecnica di *reinforcement learning* prevede che il modello operi autonomamente delle azioni, ricevendo informazioni sulle condizioni presenti del sistema e sulle conseguenze delle sue azioni su quest'ultimo, imparando la strategia migliore sulla base di un meccanismo premiale (ovverosia, riceve dei *reward* positivi o negativi in conseguenza di ogni azione intrapresa). In generale, la costruzione di un modello basato su metodi di *supervised machine learning* prevede diverse fasi:

- raccolta di tutti i dati relativi al problema di esame e costruzione del *dataset*;
- suddivisione del *dataset* nei sottoinsiemi di:
  - *training*, utilizzato per calibrare i parametri del modello;
  - *validation*, utilizzato per la valutazione delle prestazioni del modello nel corso dell'allenamento di questo, in modo da verificare che il *fitting* dei parametri proceda in modo corretto;
  - *test*, necessario per la verifica della capacità predittiva del modello una volta ultimato;
- selezione dei dati in uscita di interesse e delle *features* ritenute maggiormente significative e correlate a questi ultimi;
- pretrattamento di *features* e *labels*, procedura che consiste generalmente nell'individuazione e successiva eliminazione di *samples* anomali e nella standardizzazione dei dati;
- allenamento del modello sugli insiemi di dati costruiti in precedenza;
- valutazione delle prestazioni del modello secondo i criteri selezionati ed eventuale perfezionamento dello stesso.

### 3.1 Le reti neurali

Le reti neurali artificiali (ANN) sono una famiglia di algoritmi di *machine learning*, sviluppati per simulare il sistema nervoso umano nell'affrontare problemi specifici del ML, riportati nel paragrafo precedente. Le unità computazionali in questo modello di apprendimento sono trattate in maniera in qualche modo analoga ai neuroni biologici. Una loro peculiarità è quella di essere teoricamente in grado di apprendere qualsiasi funzione matematica, provvisto che la rete abbia un livello di complessità sufficiente (almeno un *hidden layer*) e che i dati forniti per l'allenamento siano adeguati [41]. Questa considerazione, unita all'aumento della disponibilità di dati e di potenza computazionale, ne ha determinato il successo. Lo svantaggio principale di questi metodi è la difficile interpretazione delle predizioni che generano. Essi infatti, nonostante la loro grande capacità descrittiva, non forniscono informazioni chiare sulle relazioni che costruiscono, e di conseguenza non facilitano la comprensione dei meccanismi che governano il problema a cui sono applicati.

Così come il corrispettivo biologico, una rete neurale è costituita da una moltitudine di unità computazionali interconnesse, secondo un approccio modulare; un'ulteriore analogia è il fatto che le unità computazionali sono connesse tra loro attraverso dei valori numerici, detti pesi, che rivestono un ruolo simile alla forza delle connessioni sinaptiche tra i neuroni, e l'apprendimento avviene attraverso la variazione del valore di questi pesi. L'unità fondamentale costituente la rete neurale è il neurone artificiale (Fig. 3.1). Esso riceve un determinato numero  $n$  di valori in ingresso  $x_i$  dal cosiddetto *input layer*, ne fa una somma pesata attraverso il vettore dei pesi  $\mathbf{w}$  sommando anche un termine costante  $b$ , detto *bias*, e infine fornisce un valore in uscita applicando al risultato della sommatoria una funzione  $f$ , detta di attivazione. Complessivamente si può scrivere la seguente equazione:

$$\hat{y} = f \left( \sum_{i=1}^n (w_i x_i) + b \right) = f \left( \mathbf{w}^T \mathbf{x} + b \right). \quad (3.1)$$

Una struttura formata da uno o più neuroni artificiali posti in parallelo (aventi lo stesso input), è detta perceptrone. In essa, ogni neurone possiede il suo set di pesi e applica la stessa funzione di attivazione ma fornisce un output diverso. In Eq. (3.2) è riportato lo schema di una rete neurale avente un *input layer*, costituito da un neurone per ogni *feature*, il quale ha il compito di ricevere i componenti del vettore  $\mathbf{x}$  e trasmetterli ai neuroni del *layer* successivo, i quali forniscono un valore in uscita ciascuno e lo trasmettono all'*output layer*, che ha tanti neuroni quanti sono i *labels* selezionati, e che ha il compito di fornire le risposte finali del modello.

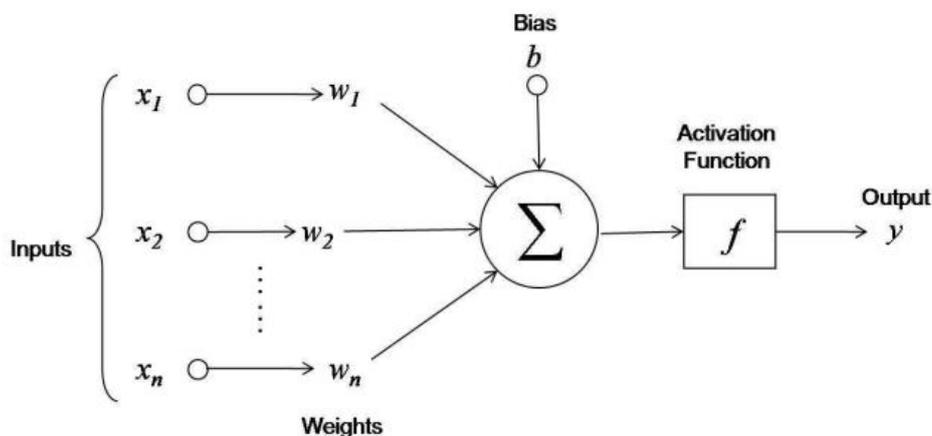


Figura 3.1: Schema di un neurone artificiale.

A partire dalla struttura del perceptrone, è possibile costruire reti più complesse interponendo tra input e l'output due o più *hidden layers*, ovvero gruppi di neuroni in parallelo, connessi in serie con il *layer* precedente e con quello successivo, in modo tale che ogni neurone riceva input da tutti i neuroni del *layer* precedente e invii il suo output a tutti i neuroni del successivo. Tali architetture sono dette reti neurali profonde (*deep neural networks*, anche chiamate *multilayer perceptrons*), di cui si riporta un esempio in Fig.

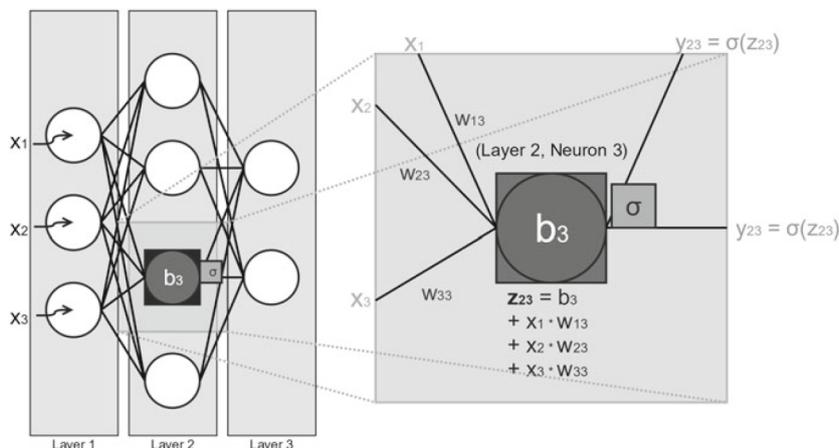


Figura 3.2: Rappresentazione schematica di un perceptrone, in dettaglio le operazioni compiute da un singolo neurone.

3.3, ed estendono la capacità del perceptrone, rendendolo quindi in grado di affrontare problemi di complessità superiore. La cosiddetta fase di *training* consiste nell'aggiornare

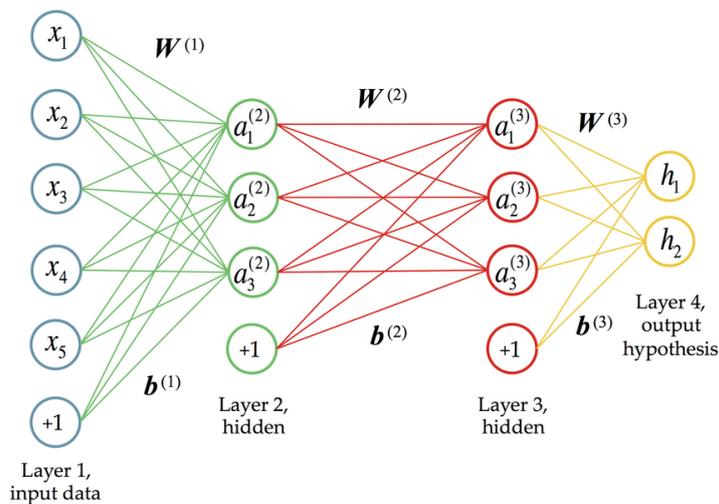


Figura 3.3: Rappresentazione schematica di una rete neurale profonda [42]. I nodi in basso, rappresentati con "+1", rappresentano il *bias* di ciascun *layer*.

iterativamente pesi e *biases* di ogni neurone in modo tale da minimizzare l'errore di predizione sul *train dataset*, utilizzando un algoritmo detto di retropropagazione. Una volta definita l'architettura della rete il processo di *training* consiste quindi di:

1. assunzione arbitraria dei valori di pesi e *biases* iniziali;
2. calcolo della predizione per ogni *sample* contenuto nell'insieme di *training*;

3. valutazione degli errori sulle predizioni effettuate;
4. valutazione del contributo di ciascun neurone agli errori sulla predizione;
5. iterazione dei passi 2,3 e 4 fino al raggiungimento di un criterio stabilito.

Nel caso di problemi di regressione, l'errore sulle predizioni viene valutato attraverso una funzione di costo, la quale valuta l'errore sulle predizioni in termini di errore assoluto medio (MAE), errore quadratico medio (MSE) o i loro logaritmi. Generalmente, la funzione di ottimizzazione (ovverosia la funzione su cui si basa l'algoritmo di retropropagazione, responsabile del *training* del modello) prevede il calcolo delle derivate parziali della funzione di costo nello spazio dei pesi e dei *biases*. Questo metodo viene chiamato *stochastic gradient descent* (SGD), il quale ad ogni iterazione calcola il gradiente locale (nell'intorno del valore corrente dei parametri) della funzione di perdita e aggiorna i pesi secondo l'equazione:

$$\mathbf{W}_{new} = \mathbf{W}_{old} + (-1)\lambda\nabla_w f_{loss} \quad (3.2)$$

dove  $f_{loss}$  è la funzione di perdita, mentre  $\lambda$  è un parametro definito dall'utente, denominato *learning rate*. In questo modo, i parametri vengono aggiornati spostandosi contro gradiente, nell'intenzione di minimizzare l'errore di predizione sul *training set*. Per quanto questo metodo non dia garanzie sul raggiungimento di un punto di minimo locale di  $f_{loss}$ , nella maggior parte dei casi permette di ridurre la funzione di costo ad un livello accettabile [43]. Il *learning rate* è uno dei cosiddetti iperparametri, ovverosia l'insieme degli aspetti della rete che vengono stabiliti a priori dall'utente, o variano secondo criteri da esso prestabiliti. I principali sono:

- numero di *hidden layers*;
- numero di neuroni di ogni *layer*;
- funzione di attivazione di ogni *layer*;
- funzione di perdita;
- funzione di ottimizzazione;
- criterio di interruzione dell'allenamento;
- *learning rate*;
- altri parametri legati all'aggiornamento dei pesi.

Il numero di *layers* e di neuroni per *layer* determina la capacità del modello e il numero di parametri che lo descrivono, di conseguenza ha un impatto determinante sul costo computazionale. In Fig. 3.4 sono rappresentate le principali funzioni di attivazione. Il *learning rate* controlla l'entità dell'aggiornamento dei pesi ad ogni iterazione. Un *learning rate* elevato permette di minimizzare velocemente la funzione di perdita ma può portare ad instabilità e ad andamenti oscillatori della funzione tra un'iterazione e l'altra, mentre valori bassi determinano una convergenza più stabile ma più lenta. Un esempio grafico del ruolo di questo parametro si può vedere in Fig. 3.5, dove si nota l'effetto positivo dell'adozione di un *learning rate* inferiore sia in termini di stabilità che di convergenza nella minimizzazione di una funzione  $f(x, y) = x^2 + ay^2$ , con  $a > 1$ .

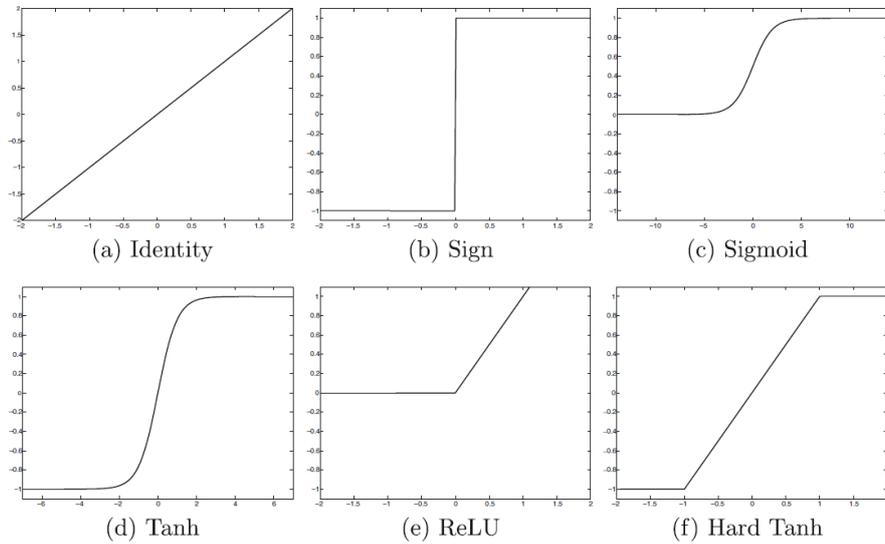


Figura 3.4: Principali funzioni di attivazione.

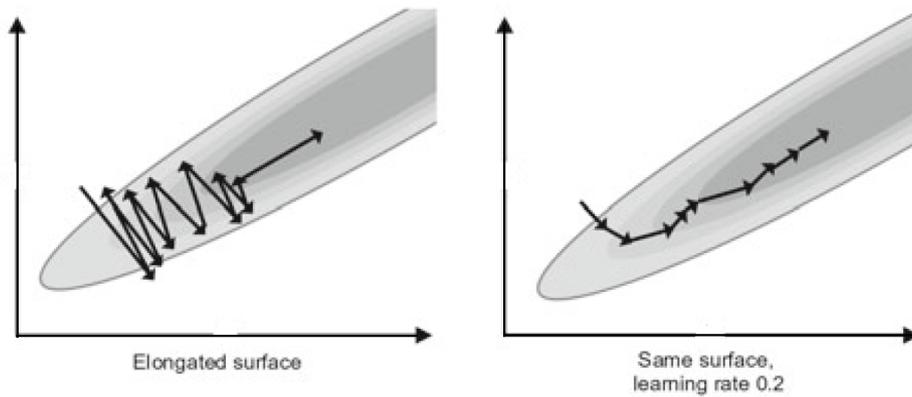


Figura 3.5: Rappresentazione grafica dell'effetto del *learning rate* nell'ottimizzazione di una funzione del tipo  $z = x^2 + ay^2$ , con  $a > 1$  [44]. Ogni vettore che compone la spezzata rappresenta un'iterazione.

## 4. Impostazione delle simulazioni e costruzione del modello

In questo capitolo viene presentata la procedura di costruzione delle simulazioni, l'estrazione dei dati di interesse e la costruzione della rete neurale funzionale alla predizione di questi. In particolare, verranno discusse la costruzione della geometria delle porzioni di letto impaccato, condotte attraverso il *software* **Blender** e l'implementazione del *bridges method* per la modifica locale dei punti di contatto attraverso il *software* **Salome**. In seguito, vengono presentate nel dettaglio le simulazioni CFD effettuate utilizzando **OpenFOAM**, i criteri di selezione delle variabili di interesse e i relativi intervalli, nonché le quantità selezionate ai fini della costruzione del modello predittivo. Infine, viene discussa l'implementazione della rete neurale, costruita attraverso la libreria **TensorFlow**.

### 4.1 Costruzione della geometria

Allo scopo di simulare la disposizione casuale delle *beads* di resina cromatografica nel *bulk* della colonna, è stata implementata una procedura *in silico* di generazione della geometria, basata sulla risoluzione delle equazioni della dinamica newtoniana relative ad un insieme di sfere che cadono, mosse dalla forza di gravità, all'interno di un cilindro cavo. Il cilindro è chiuso all'estremità inferiore e sormontato da un tronco di cono avente funzione di imbuto. Nell'ambito della simulazione, le interazioni tra gli oggetti che vi partecipano sono state simulate attraverso il modello di corpo rigido, di conseguenza le superfici coinvolte (le pareti del cilindro, il suo fondo ed il tronco di cono), come anche le sfere, sono state assunte indeformabili. Mentre le prime sono state considerate corpi statici, per le sfere è stato calcolato il comportamento dinamico attraverso il metodo presentato di seguito. Un corpo rigido è una collezione di masse puntiformi, soggette al vincolo che la distanza tra ogni paio di masse costituenti il corpo rimanga invariata. Esso possiede massa  $M$  e tensore del momento di inerzia  $\mathbf{I}$ , funzione della distribuzione spaziale delle masse puntiformi. La posizione del corpo  $i$ -esimo nel sistema di riferimento inerziale è data dal vettore posizione  $\mathbf{c}_i$ , riferito al centro di massa. In un determinato istante, un corpo rigido viene definito dalle seguenti quantità:

- posizione del centro di massa  $\mathbf{c}(t)$ ;
- orientazione (non applicabile nel caso di una sfera);
- velocità traslazionale del centro di massa  $\mathbf{v}(t)$ ;
- velocità rotazionale  $\boldsymbol{\omega}(t)$ .

Per risolvere le equazioni differenziali del moto, le forze e i momenti torcenti vengono integrati per calcolare quantità di moto e momento angolare. Queste due quantità vengono preferite alle rispettive velocità poiché sono quantità conservative. Nelle simulazioni

di corpo rigido, insieme alle equazioni di Newton-Eulero, che descrivono la dinamica dei singoli oggetti, vengono considerate tre tipologie di condizioni:

- condizione di non penetrazione, che impedisce la sovrapposizione degli oggetti;
- un modello che descrive l'attrito tra corpi a contatto;
- vincoli di complementarità, quali ad esempio le leggi che descrivono gli urti tra oggetti.

Il sistema di equazioni differenziali generato considerando tutte le condizioni precedenti viene discretizzato nel tempo e successivamente risolto algebricamente secondo una procedura iterativa, utilizzando il metodo di Eulero semi-implicito [45]. Generalmente, l'algoritmo risolutivo esegue ad ogni iterazione i seguenti compiti: la rilevazione delle collisioni tra oggetti, la determinazione delle forze di contatto e il successivo aggiornamento delle posizioni e delle velocità di tutti gli oggetti del sistema. La trattazione del problema è presentata qui in forma semplificata, poiché non è necessario includere i gradi di libertà relativi all'orientazione spaziale nel caso di sfere omogenee. Per la trattazione completa, si invita il lettore a fare riferimento a [46]. La dinamica del sistema viene risolta esprimendo lo stato di questo attraverso il vettore  $\mathbf{Y}(t)$ , detto vettore di stato, che contiene le informazioni relative a posizione del centro di massa, quantità di moto e momento angolare di tutte le  $N$  sfere. Nota la massa di ogni sfera (calcolata assumendo densità costante) è possibile calcolare il rispettivo tensore del momento di inerzia come nella seguente equazione:

$$\mathbf{I}_i = \begin{bmatrix} \frac{2}{5}M_iR_i^2 & 0 & 0 \\ 0 & \frac{2}{5}M_iR_i^2 & 0 \\ 0 & 0 & \frac{2}{5}M_iR_i^2 \end{bmatrix} \quad (4.1)$$

in cui  $M$  è la massa totale della sfera e  $R$  il suo raggio esterno. Applicando i principi della dinamica, viene costruita la seguente relazione, nella quale  $\mathbf{P}(t) = M\mathbf{v}(t)$  è la quantità di moto e  $\mathbf{L}(t) = \boldsymbol{\omega}(t)\mathbf{I}$  il momento angolare:

$$\frac{\partial \mathbf{Y}(t)}{\partial t} = \frac{\partial}{\partial t} \begin{pmatrix} \mathbf{c}_1(t) \\ \dots \\ \mathbf{c}_n(t) \\ \mathbf{P}_1(t) \\ \dots \\ \mathbf{P}_n(t) \\ \mathbf{L}_1(t) \\ \dots \\ \mathbf{L}_n(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1(t) \\ \dots \\ \mathbf{v}_n(t) \\ \mathbf{F}_1(t) \\ \dots \\ \mathbf{F}_n(t) \\ \boldsymbol{\tau}_1(t) \\ \dots \\ \boldsymbol{\tau}_n(t) \end{pmatrix}. \quad (4.2)$$

Nell'ambito delle simulazioni effettuate, è stato introdotto un coefficiente, chiamato *linear damping*, per simulare in maniera semplificata l'effetto delle forze viscosse generate dal fluido in cui sono immerse le particelle. Questo coefficiente rappresenta la decrescita della velocità lineare dell'oggetto, in punti percentuali, nell'arco temporale di un secondo.

Trattandosi di una simulazione di corpo rigido, si rende necessario evitare l'interpenetrazione tra oggetti. In questo senso, la superficie delle sfere è la loro superficie geometrica esatta, mentre gli altri oggetti (cilindro, tronco di cono e piano), sono stati approssimati attraverso *mesh* triangolari. Poiché la collisione avviene istantaneamente, la discontinuità nelle velocità degli oggetti coinvolti rende necessario introdurre una modifica all'algoritmo risolutivo delle equazioni differenziali ordinarie riportate in Eq. (4.2), che consiste nel descrivere le collisioni attraverso leggi di contatto invece che leggi di forza. Il metodo integra le equazioni della dinamica con due condizioni aggiuntive: la condizione di complementarità di Signorini e la legge dell'attrito di Coulomb. La prima formalizza la considerazione che se la distanza tra due corpi rigidi ( $\delta_n$ ) è positiva, la forza normale ( $f_n$ ) deve essere nulla, mentre se la distanza è zero la forza normale è necessariamente nulla se la componente normale della velocità relativa tra i due corpi è positiva (ovverosia se i corpi si stanno allontanando):

$$\begin{cases} \delta_n > 0 \Rightarrow f_n = 0 \\ \delta_n = 0 \wedge \begin{cases} u_n > 0 \Rightarrow f_n = 0 \\ u_n = 0 \Rightarrow f_n \geq 0. \end{cases} \end{cases} \quad (4.3)$$

La seconda è una relazione di complementarità tra la forza di attrito  $\mathbf{f}_t$  e la componente tangenziale della velocità relativa nel punto di contatto,  $\mathbf{u}_t$ , attraverso il coefficiente empirico  $\mu_a$  definito per superfici omogenee e isotrope:

$$\begin{cases} u_t > 0 \Rightarrow f_t = -\mu_a f_n \\ u_t = 0 \Rightarrow -\mu_a f_n \leq f_t \leq \mu_a f_n \\ u_t < 0 \Rightarrow f_t = \mu_a f_n. \end{cases} \quad (4.4)$$

A causa della discretizzazione temporale applicata, e della discontinuità nella velocità a seguito di un urto, le velocità utilizzate all'interno delle relazioni esposte sono calcolate come una media pesata tra la velocità relativa nell'istante precedente all'urto ( $\mathbf{u}^-$ ) e il suo valore nell'istante successivo ( $\mathbf{u}^+$ , incognito), secondo le relazioni riportate in Eq. (4.6), in cui i pesi  $e_n$  e  $e_t$  sono detti coefficienti di restituzione rispettivamente normale e tangenziale, definiti secondo le equazioni:

$$e_n = -\frac{u_n^+}{u_n^-}, \quad e_t = -\frac{u_t^+}{u_t^-}, \quad (4.5)$$

$$u_n = \frac{u_n^+ + e_n u_n^-}{1 + e_n}, \quad u_t = \frac{u_t^+ + e_t u_t^-}{1 + e_t}. \quad (4.6)$$

Poiché si desidera che i corpi modifichino istantaneamente la loro velocità, si introduce una nuova quantità, l'impulso, definito come  $\mathbf{J} = \mathbf{F}\Delta t$ , dove  $\Delta t$  indica il tempo finito di applicazione della forza  $\mathbf{F}$ . Questa quantità si relaziona con le componenti traslazionali e rotazionali della velocità della sfera attraverso le relazioni seguenti, dove  $R$  è il raggio della sfera.

$$\Delta \mathbf{v}_{rel} = \frac{\mathbf{J}_n}{M} \quad \Delta \boldsymbol{\omega}_{rel} = \mathbf{I}^{-1} R \mathbf{J}_t \quad (4.7)$$

Sostituendo le Eq. (4.4) e (4.6) in Eq. (4.7), è possibile risolvere il sistema in  $\mathbf{J}$ . Una volta calcolato l'impulso generato dalla collisione, questo viene inserito nelle equazioni della dinamica, integrate su  $\Delta t$ , in modo da ottenere le equazioni:

$$M(\mathbf{u}^+ - \mathbf{u}^-) = \mathbf{J}_n + \Delta t \mathbf{F}_{ext} \quad (4.8)$$

$$\mathbf{I}(\omega^+ - \omega^-) = \mathbf{J}_t R + \Delta t \boldsymbol{\tau}_{ext} \quad (4.9)$$

Il lettore interessato può trovare la trattazione matematica del problema, semplificata nel caso senza attrito, in [47] e [48].

#### 4.1.1 Geometria del contenitore

La creazione della geometria è stata condotta attraverso il software **Blender**, il quale fa uso della libreria *bullet physics* [49]. La geometria del contenitore delle sfere, rappresentata in Fig. 4.1, è stata costruita in modo tale da garantire una distanza minima tra la parete del contenitore e gli estremi del volume estratto per la successiva simulazione CFD, il quale è un parallelepipedo con gli spigoli paralleli agli assi cartesiani. Ciò è stato necessario per ridurre gli effetti dovuti alla presenza delle pareti nella distribuzione spaziale nelle sfere, così facendo si ottiene un risultato maggiormente rappresentativo del *bulk* del letto impaccato, in particolare un grado di vuoto del letto maggiormente omogeneo. In ogni simulazione il fondo del cilindro dista almeno tre diametri medi dalla superficie inferiore del volume estratto, mentre la superficie cilindro dista almeno un diametro medio dagli spigoli verticali del parallelepipedo (Fig. 4.2).

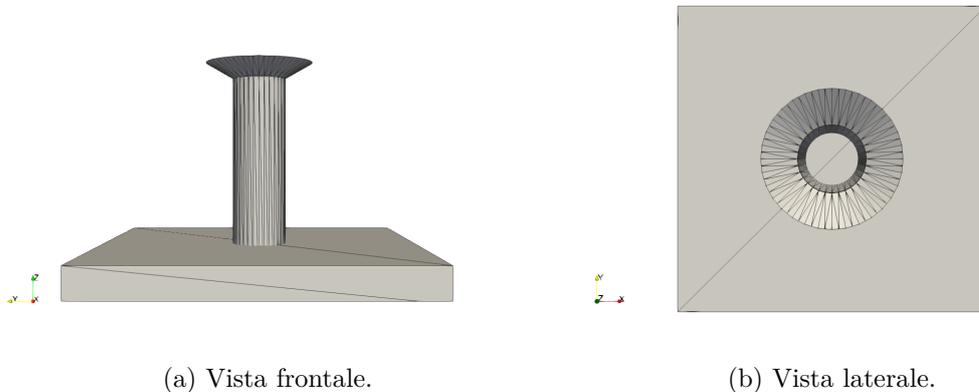
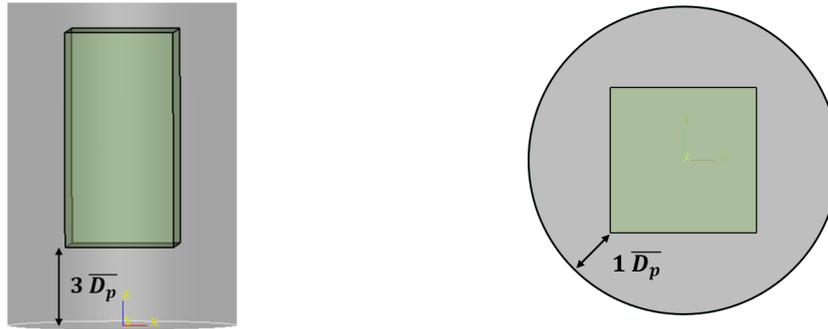


Figura 4.1: Vista frontale e dall'alto del cilindro di contenimento delle sfere, del piano sottostante e del tronco di cono sovrastante.

#### 4.1.2 Disposizione iniziale delle sfere

All'istante iniziale le sfere sono disposte verticalmente al di sopra del cilindro, organizzate in 100 piani lungo l'asse  $z$ , distanziati tra loro di 1.5 diametri medi. Ogni piano è strutturato come una griglia  $4 \times 4$ , perpendicolare e simmetrica rispetto all'asse  $z$ , i cui



(a) distanza minima dal fondo.

(b) distanza minima dalla parete cilindrica.

Figura 4.2: Distanze minime considerate tra le pareti del contenitore e il volume estratto per la simulazione.

nodi distano 1.5 diametri medi lungo x e lungo y, per un totale di 1600 sfere. I centri delle sfere sono stati inizialmente collocati in corrispondenza dei nodi della griglia, successivamente è stata applicata a questi una traslazione casuale lungo una direzione arbitraria, e di modulo compreso tra zero e il raggio medio delle sfere. Questo accorgimento è stato implementato per evitare la tendenza delle sfere a disporsi lungo direzioni preferenziali, producendo colonne impilate.

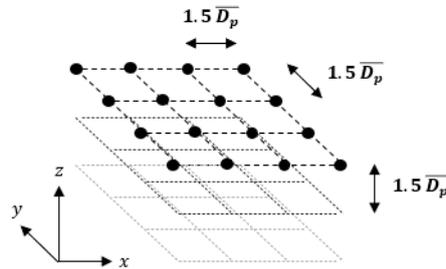


Figura 4.3: Rappresentazione schematica della griglia di disposizione iniziale delle sfere.

### 4.1.3 Distribuzione dei diametri delle sfere

Per poter meglio approssimare la geometria di un reale letto cromatografico, si è reso necessario produrre una distribuzione dei diametri delle sfere, su base volumetrica. La polidispersità delle particelle di resina è stata approssimata attraverso una distribuzione normale, centrata sulla media della distribuzione volumetrica dei diametri ( $d_{v50}$ ), troncata agli estremi dell'intervallo di valore del raggio considerato ( $12 - 82.5 \mu m$ ). I valori discreti dei raggi sono stati generati automaticamente, in accordo con la densità di probabilità selezionata, e raccolti in un vettore. Successivamente, i valori sono stati disposti

all'interno del vettore in un ordine casuale, e assegnati alle sfere in maniera sequenziale. Un esempio di distribuzione generata in questo modo è riportato in Fig. 4.4.

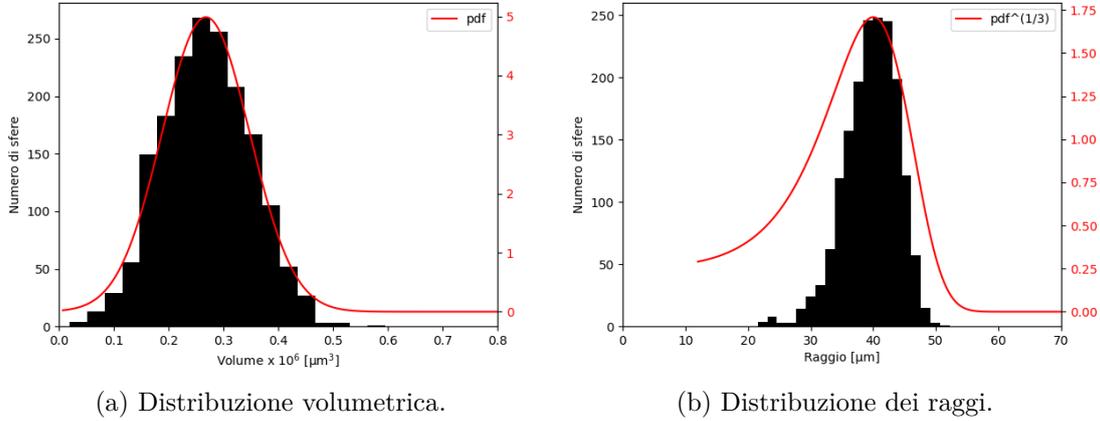


Figura 4.4: Esempio di distribuzione dimensionale delle sfere generata all'interno della simulazione, avente diametro medio pari a  $40 \mu m$  e deviazione standard pari a  $0.08 \times 10^6 \mu m^3$ . Le curve riportate in rosso sono rispettivamente la funzione densità di probabilità e la sua radice cubica.

#### 4.1.4 Condizioni di simulazione

Le simulazioni sono state condotte utilizzando i parametri riassunti in Tab. 4.1. Il processo reale di impaccamento di una colonna cromatografica è significativamente diverso dal processo simulato, poiché nel primo caso le particelle di resina vengono caricate nella colonna trasportate da un flusso di liquido, e la procedura è controllata attraverso la differenza pressione applicata a cavallo della colonna. Considerata la complessità di simulazione di un sistema del genere, il presente modello non ambisce a descrivere accuratamente la procedura quanto piuttosto a generare una geometria sufficientemente simile (in termini di porosità del letto e distribuzione dimensionale dei canali). I parametri sono quindi stati scelti in modo arbitrario, osservando in seguito che i valori scelti forniscono geometrie con grado di vuoto aderente a quello atteso in colonne cromatografiche delle tipologie considerate nel presente lavoro, specificate in seguito. In questo senso, non è stata effettuata una validazione completa della metodologia di simulazione, e un'indagine approfondita sull'ottimizzazione dei parametri presentati in tabella è auspicabile. La simulazione restituisce un file testuale (CSV, *Comma Separated Values* che raccoglie raggio e coordinate del centro di ogni sfera al termine della simulazione. Lo *script*, in linguaggio Python, utilizzato per generare i raggi delle sfere, disporle nella posizione iniziale ed effettuare la simulazione è riportato in appendice A.

Tabella 4.1: Impostazioni utilizzate per la simulazione in Blender.

Parametro	Valore
Accelerazione gravitazionale	9.8 $m/s^2$
Densità delle sfere	1.0 $kg/m^3$
<i>Collision shape</i> sfere	<i>Sphere</i>
<i>Collision shape</i> altri elementi	<i>Mesh</i>
<i>Collision margin</i>	Non utilizzato
Coefficiente di attrito sfere	0.5
Coefficiente di attrito altri elementi	0.5
<i>Linear damping</i>	0.1
<i>Angular damping</i>	0
Restituzione normale	0.25
Restituzione tangenziale	0.25
<i>Timesteps</i> al secondo	60
Iterazioni per <i>step</i>	10
Tempo di simulazione	250 s

#### 4.1.5 Modellazione dei punti di contatto

I punti di contatto tra le sfere sono stati oggetto di modifica locale attraverso il *bridges method*. La procedura automatica di generazione dei *bridges* tra le sfere è stata implementata utilizzando il *software Salome*. In particolare, si è utilizzato uno *script* in linguaggio Python, riportato in appendice B, avente le seguenti funzioni:

- importare raggio e posizione delle sfere risultanti dalla simulazione di impaccamento, considerando esclusivamente le sfere del tutto o parzialmente contenute nel volume simulato;
- individuare i punti di contatto tra sfera e sfera;
- generare in corrispondenza dei punti di contatto un cilindro di raggio prestabilito, con asse passante per i centri delle due sfere in contatto.

Nel procedimento, due sfere sono definite a contatto se la distanza tra i rispettivi centri, al netto della somma dei raggi, risulta minore di un valore arbitrario, fissato a  $0.1 \mu m$ , come riportato in Eq. (4.10). Questo valore è necessario per limitare gli effetti di errori numerici di piccola entità che possono verificarsi nella simulazione di impaccamento, che generano punti di contatto in cui la distanza sfera-sfera non corrisponde esattamente a 0. Nell'equazione,  $d_{i,j}$  rappresenta la distanza tra i centri delle sfere,  $\mathbf{c}$  la posizione del centro e  $R$  il raggio.

$$d_{i,j} = |\mathbf{c}_i - \mathbf{c}_j| - (R_i + R_j) < d_c = 0.1 \mu m. \quad (4.10)$$

Il punto di contatto è definito come il punto medio del segmento  $d$ . Il raggio dei cilindri ponte non dipende dai raggi delle sfere a contatto ma assume un unico valore per tutti

i punti di contatto, in rapporto al diametro medio delle particelle. In particolare, il rapporto tra raggio dei cilindri ( $r_b$ ) e diametro medio delle sfere ( $\bar{D}_p$ ) è fissato a 0.05, pari alla metà del rapporto massimo consigliato in [35]. Dopo aver individuato il punto di contatto e la direzione dell'asse passante tra i centri delle sfere, da ognuna delle due viene rimossa una calotta sferica perpendicolare all'asse e di altezza pari a:

$$h_{cs} = R_i \left[ 1 - \cos \left( \sin^{-1} \left( \frac{r_b}{R_i} \right) \right) \right] + 0.01R_i \quad (4.11)$$

Questo passaggio (in particolare il termine  $0.01R_i$ ) si è reso necessario per evitare di incorrere in errori nella successiva fase di “fusione” dei solidi in un unico oggetto. In seguito viene interposto tra le due sfere il cilindro di raggio  $r_b$  e viene tagliato alle estremità dalle superfici piane generate dalla rimozione della calotta sferica. Il procedimento è schematizzato in Fig. 4.5. Una volta ultimato il trattamento per tutti i punti di contatto presenti nel volume considerato, si genera un unico solido (e quindi un'unica superficie) attraverso un'operazione di fusione di tutti gli oggetti adiacenti. Dopodiché, le informazioni relative alla superficie del solido risultante vengono esportate per poter essere elaborate dal *mesher* (Fig. 4.6).

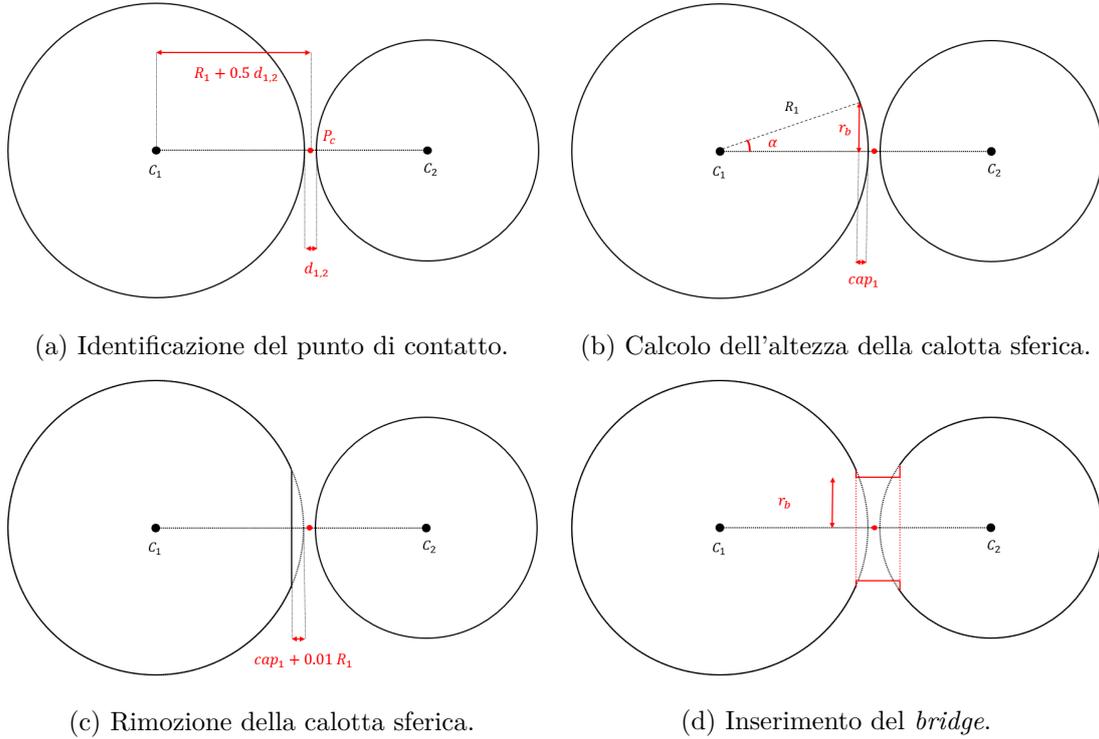
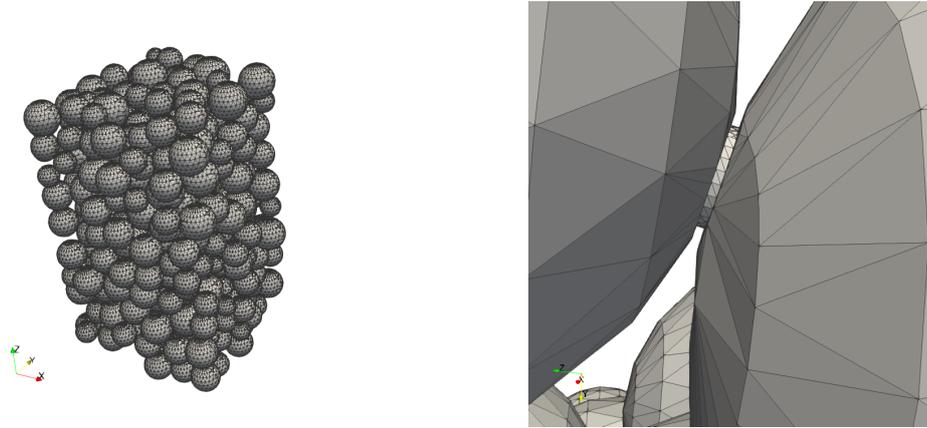


Figura 4.5: Schematizzazione della procedura di implementazione del *bridges method*.



(a) Esempio di geometria esportata da Salome.

(b) Dettaglio del *bridge* posto in corrispondenza di un punto di contatto.

Figura 4.6: Esempio di una geometria esportata in formato STL in seguito al trattamento dei punti di contatto.

## 4.2 Impostazione delle simulazioni fluidodinamiche

In questa sezione viene descritta in dettaglio la preparazione delle simulazioni fluidodinamiche effettuate per la costruzione del *dataset*. Le simulazioni sono state effettuate allo scopo di ottenere i valori di:

- porosità esterna del letto ( $\epsilon$ );
- area interfacciale specifica tra dominio fluido e *beads* di resina ( $a_{fs}$ );
- permeabilità del letto ( $k_{bed}$ ), espressa attraverso il suo inverso, il coefficiente di Darcy ( $d = \frac{1}{k_{bed}}$ );
- coefficiente di scambio termico medio ( $h_{fs}$ ).

A questo scopo sono state effettuate due diverse simulazioni in serie su ogni geometria generata come esposto in precedenza. Dal processo di costruzione della griglia computazionale sono stati estratti i valori di volume dei domini rispettivamente di fluido e di resina ( $V_f, V_s$ ) e l'area totale dell'interfaccia fluido-resina ( $A$ ). Utilizzando questi dati sono state calcolate la porosità esterna del letto e l'area interfacciale specifica, secondo le equazioni:

$$\epsilon = \frac{V_f}{V_f + V_s}, \quad a_{fs} = \frac{A}{V_f + V_s}. \quad (4.12)$$

Una volta costruita la griglia, è stata eseguita una simulazione sul solo volume di fluido allo scopo di calcolare il campo di moto, allo stato stazionario, generato da una differenza di pressione applicata agli estremi del dominio. In Fig. 4.7 è riportato un esempio dei campi di velocità e pressione calcolati lungo una sezione parallela all'asse  $z$ . Utilizzando

il valore di velocità, mediato sul volume di fluido, in direzione parallela al gradiente di pressione ( $\bar{v}_z$ ) applicato, si è determinato il coefficiente di Darcy relativo al moto del fluido nella direzione determinata secondo l'Eq (4.13), ovvero sia l'equazione di Darcy in forma integrale. L'utilizzo dell'Eq. (4.13), la quale non contiene termini inerziali ma è esclusivamente funzione lineare della velocità superficiale del fluido ( $v_{sup}$ ), è giustificato dal fatto che il numero di Reynolds (riferito alle particelle,  $Re_p$ ) è significativamente inferiore all'unità, indicando una condizione di moto completamente laminare. In particolare,  $Re_p = O(10^{-2})$ .

$$d = \frac{\Delta p}{L} \frac{1}{\mu u_{sup}} = \frac{\Delta p}{L} \frac{1}{\mu \bar{u}_z \epsilon}. \quad (4.13)$$

Una volta calcolato il campo di moto, questo è stato importato in una simulazione multifase transitoria, nella quale sono state risolte anche le equazioni dell'energia (Eq. (2.3) e (2.4)), allo scopo di determinare le condizioni di trasporto del calore all'interno della fase fluida e della resina, e lo scambio della quantità tra i due domini. In particolare, si è calcolato il flusso di calore a cavallo dell'interfaccia fluido-resina e le temperature medie nei due domini in ogni istante.

#### 4.2.1 Geometria e griglia di calcolo

Il volume selezionato per la simulazione è un parallelepipedo a sezione quadrata, avente la base perpendicolare all'asse z e altezza pari al doppio del lato di base. La geometria è costruita estraendo le informazioni relative alla superficie dell'impaccamento, provenienti dal `Salome`, nella zona interna al parallelepipedo. La posizione di quest'ultimo è stabilita secondo i criteri esposti in precedenza e riassunti in Fig. 4.2. Le dimensioni della cella base e del volume considerato per la simulazione sono espresse in seguito (sezione 4.4). Una volta generata la griglia base attraverso la `utility blockMesh`, la griglia è stata raffinata e adattata alla geometria dell'impaccamento attraverso `snappyHexMesh`.

#### 4.2.2 Proprietà dei materiali e condizioni al contorno

Le proprietà termofisiche del fluido considerato all'interno delle simulazioni sono approssimativamente quelle dell'acqua a 20 °C. Per quanto riguarda la resina, considerata l'elevata porosità interna che la contraddistingue, densità e calore specifico sono stati considerati pari a quelli del fluido, e solamente la conducibilità termica è stata variata all'interno dell'intervallo definito in seguito. I valori utilizzati sono riportati in Tab. 4.2. Le facce

Tabella 4.2: Proprietà termofisiche dei materiali.

Proprietà	Fluido	Resina	Unità di misura
Densità ( $\rho$ )	998.2	998.2	$kg/m^3$
viscosità dinamica ( $\mu$ )	1	-	$Pa \cdot s$
Calore specifico ( $C_p$ )	4184.2	4184.2	$J/(kg \cdot K)$
Conducibilità termica ( $k$ )	0.6	variabile	$W/(m \cdot K)$

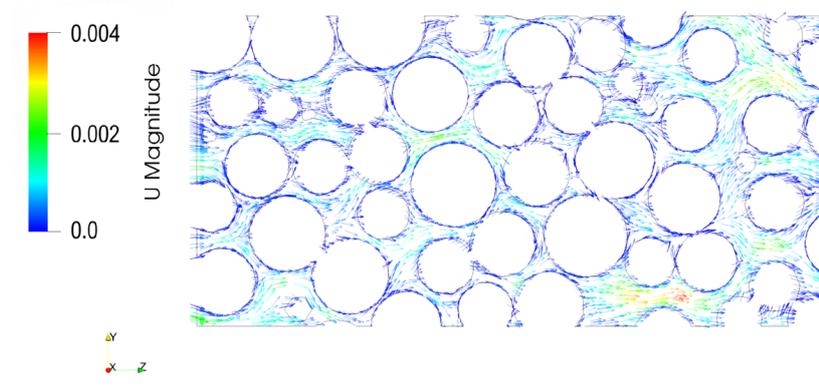
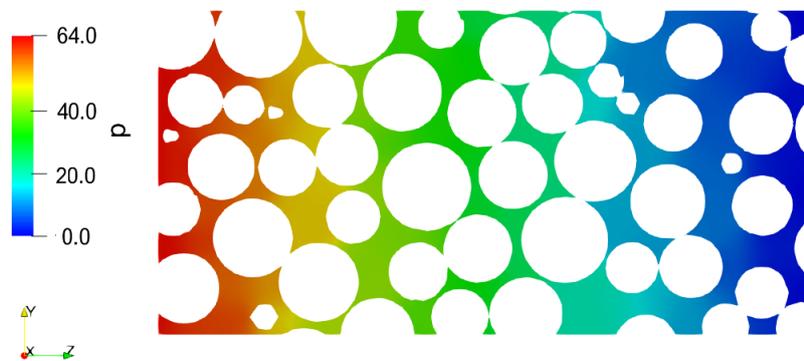
(a) Campo di velocità ( $m/s$ ).(b) Campo di pressione ( $Pa$ ).

Figura 4.7: Esempio dei campi di velocità e pressione calcolati lungo una sezione parallela all'asse  $z$ .

del parallelepipedo simulato sono state raggruppate in tre insiemi: ingresso, uscita e facce laterali (Fig. 4.8, 4.9, 4.10). Sia nella simulazione stazionaria che in quella transitoria è stata imposta una condizione di simmetria alle facce laterali. Ciò corrisponde all'imporre nullo il flusso di qualsiasi quantità attraverso la superficie. Questa è la condizione che maggiormente si avvicina alle condizioni reali, in cui il volume considerato è circondato in ogni direzione da altri volumi a lui simili, e il flusso principale di fluido ha direzione parallela alle facce considerate. Nella simulazione del campo di moto, solo il dominio fluido è considerato, e viene risolto esclusivamente il campo di moto. A tale scopo, sulle superfici di ingresso e uscita vengono definiti dei valori di pressione, uniformi, tali da generare la differenza di pressione desiderata tra gli estremi del dominio; per la velocità è stata invece imposta la condizione `zeroGradient`, in modo tale da permettere il calcolo di un profilo di velocità all'ingresso già sviluppato, derivato dal campo di pressione. In corrispondenza della superficie delle sfere di resina è stata imposta una condizione di `noSlip` alla velocità e di `zeroGradient` alla pressione. Nella simulazione transitorie, sono

state mantenute le stesse condizioni di velocità e pressione nel dominio fluido, nonché lo stesso campo di moto, alla quale sono state aggiunte le condizioni al contorno relative all'equazione dell'energia. Sulla faccia di ingresso (sia nella porzione fluida che di resina) viene imposto un valore uniforme di temperatura, pari a  $19.5\text{ }^{\circ}\text{C}$ ; all'istante iniziale l'intero sistema si trova alla temperatura uniforme di  $25\text{ }^{\circ}\text{C}$ , al progredire della simulazione il sistema si raffredda avvicinandosi asintoticamente alla temperatura della superficie di ingresso.

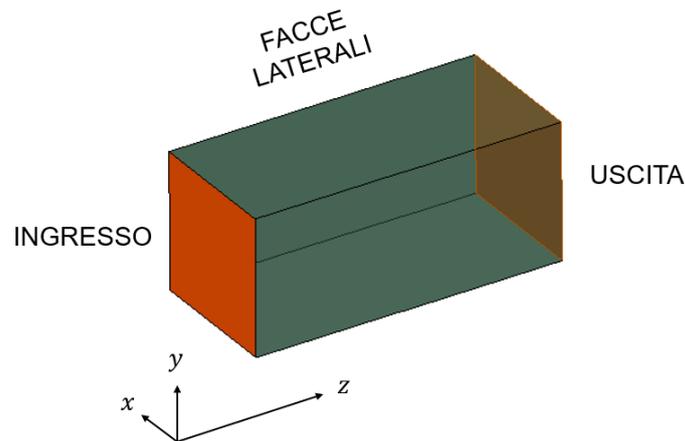


Figura 4.8: Rappresentazione schematica del dominio di simulazione.

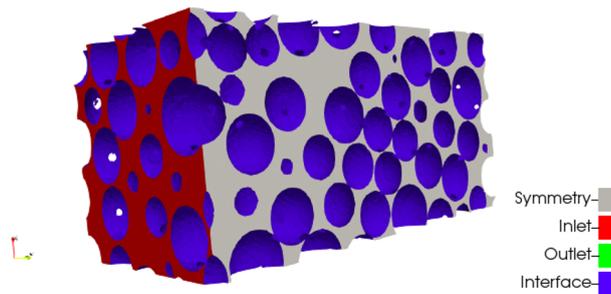


Figura 4.9: Esempio di dominio fluido simulato. Viene riportata la relativa suddivisione delle facce del parallelepipedo.

Le simulazioni sono state condotte attraverso una parallelizzazione delle stesse in 12 sottoprocessi, eseguiti in un cluster provvisto di nodi con CPU Intel Skylake 6154 (3.0 GHz, 36-cores, 384 GB RAM) e Intel Cascade Lake 6240 (2.6 GHz, 36-cores, 2 TB RAM).

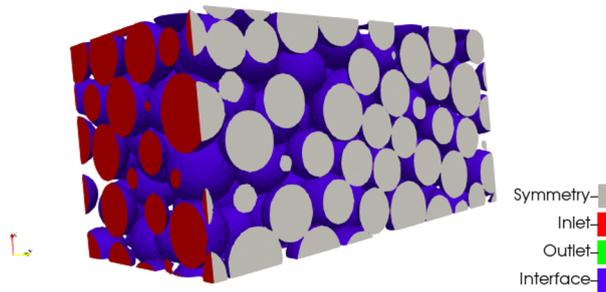


Figura 4.10: Esempio di dominio resina simulato. Viene riportata la relativa suddivisione delle facce del parallelepipedo.

### 4.2.3 Schemi numerici e criteri di convergenza adottati

L'algoritmo risolutivo scelto per il campo di moto e il *pressure-velocity coupling* nella simulazione in stato stazionario è `simpleFoam`, l'implementazione nativa di `OpenFOAM` dello schema SIMPLE, applicando `nCorrectors` pari a 2 e `nNonOrthogonalCorrectors` pari a 2. La simulazione transitoria viene condotta utilizzando il solver `chtMultiRegionFoam`, sviluppato per applicazioni a problemi di scambio termico in regioni multiple accoppiate, sia fluide che solide. Quest'ultimo è stato scelto per la sua flessibilità, poiché offre la possibilità di introdurre funzioni di stato che descrivano la variazione delle proprietà termofisiche dei materiali al variare della temperatura. In ogni caso, nello studio presente queste proprietà sono state considerate costanti in luce del ridotto intervallo di temperatura sperimentato dal sistema. In conseguenza a ciò, nel corso della simulazione transitoria si è ritenuto non necessario risolvere le equazioni del moto, e i valori di pressione e velocità del fluido sono importati dalla simulazione in stato stazionario. L'equazione dell'energia viene risolta utilizzando il metodo di Eulero implicito per la discretizzazione temporale ed eseguendo 5 iterazioni per *timestep* (Schema PIMPLE, con `nCorrectors` pari a 1 e `nOuterCorrectors` pari a 5). In entrambe le simulazioni, per il calcolo delle divergenze è stato adottato lo schema `bounded Gauss linearUpwind`, descritto nella sezione 2.2.2, i gradienti sono stati calcolati utilizzando lo schema `Gauss linear`, mentre i laplaciani utilizzando lo schema `Gauss linear corrected`. I fattori di sottorilassamento e i criteri di convergenza applicati alle quantità calcolate sono riportati in Tab. 4.3.

## 4.3 Progettazione del dataset

Il presente studio ha come obiettivo la costruzione di un modello numerico in grado di simulare il comportamento di colonne cromatografiche industriali per processi di separazione di proteine. Alla luce di ciò, il *dataset* è stato costruito in riferimento alle tipiche condizioni operative di queste ultime, e attraverso una stima delle proprietà delle resine cromatografiche utilizzate a questo proposito sulla base dei dati, forniti dai rispettivi produttori, di una collezione di 27 diverse resine utilizzate correntemente presso i laboratori

Tabella 4.3: Fattori di sottorilassamento e criteri di convergenza adottati.

Quantit�	Sottorilassamento	Tolleranza
	<i>Dominio fluido</i>	
Pressione	0.3	$10^{-6}$
Velocit�	0.4	$10^{-6}$
Entalpia	0.75	$10^{-6}$
	<i>Dominio resina</i>	
Entalpia	0.75	$10^{-6}$

senesi di GSK Vaccines. Nel corso dello studio i parametri selezionati come rilevanti nella predizione delle quantit  di interesse sono:

- diametro medio (dv50) delle particelle di resina;
- deviazione standard della distribuzione volumetrica dei diametri, approssimata attraverso una distribuzione normale;
- numero di Reynolds del sistema, riferito alle particelle ( $Re_p$ );
- diffusivit  termica della resina, espressa come valore relativo rispetto all'acqua.

In Tab. 4.4 sono riassunti gli intervalli di valori esplorati per ogni *feature* scelta. Complessivamente, sono state effettuate 130 simulazioni. Di queste, 100 costituiscono il *training set*, 15 il *validation set* e 15 il *test set*.

Tabella 4.4: Intervalli di variazione delle *features* nella costruzione del *dataset*.

<i>Feature</i>	Intervallo	Unit� di misura
Diametro medio ( $\bar{D}_p$ )	35 – 90	$\mu m$
Deviazione standard	$f(\bar{D}_p)$	$\mu m^3 \times 10^6$
Numero di Reynolds ( $Re_p$ )	$4.50 - 173.49 \times 10^{-3}$	adimensionale
Diffusivit� termica relativa della resina ( $\alpha_{rel,s}$ )	70 – 120%	adimensionale

Nel caso del diametro medio delle *beads*, si   fatto riferimento ai valori minimo e massimo di dv50, ovverosia il massimo valore di diametro al di sotto della quale si trova il 50% del volume totale del campione di particelle di resina, forniti dai produttori delle diverse resine. L'intervallo selezionato   35 – 90  $\mu m$ . A causa della indisponibilit  di dati sulla distribuzione dimensionale delle resine, la distribuzione volumetrica dei diametri   stata approssimata con una distribuzione normale; il valore di deviazione standard massimo sperimentabile per un determinato valore di diametro medio   funzione lineare di quest'ultimo. La funzione, riportata in Eq. (4.14),   stata costruita per interpolazione,

tra il valore massimo, fissato a  $0.15 \times 10^6 \mu m^3$ , corrispondente a particelle con diametro medio pari a  $90 \mu m$ , e il valore minimo, fissato a  $0.01 \times 10^6 \mu m^3$ , per particelle con diametro medio pari a  $35 \mu m$ , come riportato in Eq. (4.15):

$$f_{MAX,std}(\bar{D}_p) = 0.0025455\bar{D}_p - 0.079095, \quad (4.14)$$

$$\begin{cases} f_{MAX,std}(35\mu m) = 0.01 \times 10^6 \mu m^3 \\ f_{MAX,std}(90\mu m) = 0.15 \times 10^6 \mu m^3. \end{cases} \quad (4.15)$$

In particolare, il valore massimo di  $f_{MAX,std}(\bar{D}_p)$  è stato scelto per limitare l'inserimento

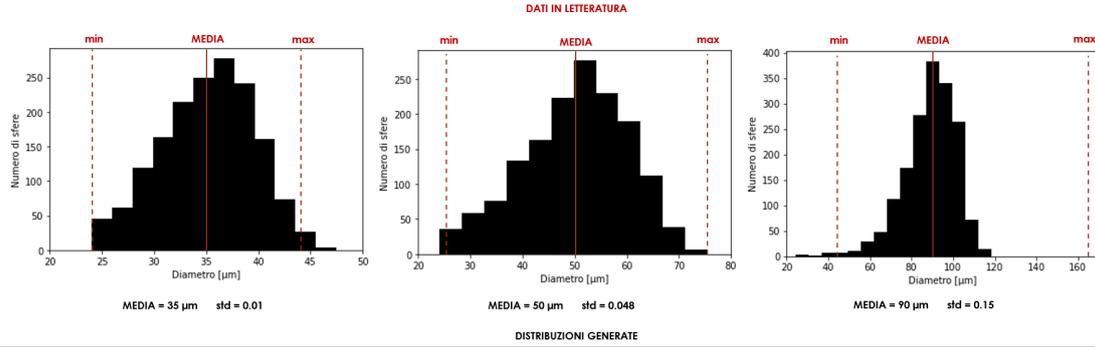


Figura 4.11: Confronto tra le distribuzioni di massima ampiezza generate attraverso l'Eq. (4.14) e gli intervalli di diametri forniti in letteratura (in rosso) per determinati valori di diametro medio.

di particelle di dimensioni eccessive nel sistema, le quali introducono disomogeneità nel letto e rendono necessario eseguire le simulazioni su volumi rappresentativi (REV) di dimensioni considerevoli; mentre il valore minimo è stato scelto in modo tale da adattare gli estremi della distribuzione ai valori di diametro minimo e massimo, forniti dal produttore, per le particelle di diametro medio pari a  $35 \mu m$ . Alcuni esempi riguardanti l'aderenza delle distribuzioni così generate ai dati forniti dal produttore è riportata in Fig. 4.11. Nella tipologia di colonne cromatografiche in esame l'intervallo operativo di velocità superficiale lungo l'asse corrisponde approssimativamente a  $30 - 200 \text{ cm/h}$  ( $8.33 - 55.56 \times 10^{-5} \text{ m/s}$ ); il dato è fornito da GSK Vaccines, può variare anche notevolmente in funzione della tipologia di resina e delle dimensioni della colonna. Nelle simulazioni effettuate la differenza di pressione applicata è tale da mantenere la velocità superficiale lungo l'asse  $z$  approssimativamente all'interno di questo intervallo. In particolare, le velocità superficiali sperimentate all'interno del *training set* vanno da  $17.00$  a  $298.61 \text{ cm/h}$ . Il parametro scelto come *feature* all'interno della rete neurale è il numero di Reynolds riferito alle particelle, definito nella seguente equazione:

$$Re_p = \frac{\rho f u_{sup} \bar{D}_p}{\mu}, \quad (4.16)$$

dove  $u_{sup}$  è la velocità superficiale del fluido lungo l'asse  $z$ . Il valore di  $Re_p$  varia all'interno dell'intervallo  $0.00488 - 0.17349$ . Infine, i valori di diffusività termica della resina sono

stati selezionati sulla base di una stima, poiché i produttori non forniscono informazioni sulle proprietà termofisiche della resina. Di conseguenza, si è proceduto individuando un valore indicativo minimo di porosità interna, pari a 65%, e un intervallo di valori di diffusività termica della fase solida vera e propria ( $2.0 - 22.0 \times 10^{-8} m^2/s$ , considerando diversi materiali polimerici e non, come agarosio, poliacrilati e idrossiapatite). Successivamente, sono stati individuati gli estremi attraverso l'interpolazione lineare delle proprietà dell'acqua e del solido, attraverso l'Equazione:

$$\alpha_{rel,s} = 0.65 + 0.35 \frac{\alpha_{solido}}{\alpha_{acqua}}. \quad (4.17)$$

## 4.4 Analisi preliminari

In questa sezione sono riportati gli studi preliminari effettuati allo scopo di selezionare le impostazioni adeguate per le simulazioni fluidodinamiche. L'accuratezza della simulazione numerica è cruciale non solo per l'attendibilità di questa, ma ha un effetto rilevante anche sulla capacità predittiva della rete neurale. *Samples* caratterizzati da un elevato errore sperimentale, numerico o modellistico possono determinare una non univocità della relazione causa-effetto tra *features* e *labels*, generando errori significativi sulla predizione di *samples* non inclusi nel *training set* (*generalization error*). In altre parole, è necessario che i dati utilizzati per l'allenamento della rete siano consistenti, cioè che l'errore sulle predizioni tenda a zero al crescere delle dimensioni del *dataset di allenamento*. A tale scopo, è stato determinato il volume di letto impaccato minimo (REV) tale da essere rappresentativo delle condizioni dell'intero letto, eseguendo una verifica sulla porosità esterna calcolata al variare del volume simulato. Successivamente, è stata determinata la dimensione della cella base in grado di garantire l'indipendenza dei risultati dal grado di infittimento della griglia, e si è determinata la discretizzazione temporale ottimale. Infine, si è proceduto a effettuare un'analisi della varianza sperimentale delle grandezze calcolate (porosità esterna, area interfacciale specifica, coefficiente di Darcy e coefficiente di scambio termico) utilizzando le impostazioni determinate in precedenza. L'analisi è stata condotta allo scopo di ottenere un riferimento per la valutazione delle prestazioni della rete neurale; in particolare per determinare se l'errore generato sulle predizioni abbia lo stesso ordine di grandezza della varianza sperimentale.

### 4.4.1 Selezione del volume di controllo

La selezione dell'elemento di volume rappresentativo è stata effettuata valutando la porosità esterna calcolata al variare delle dimensioni del volume simulato. Poiché l'obiettivo della simulazione fluidodinamica è ottenere informazioni che siano rappresentative delle condizioni dell'intero letto e non esclusivamente del volume simulato, quest'ultimo deve essere omogeneo, ovvero deve contenere un numero di sfere tale da garantire un'adeguata distribuzione dei diametri di queste al suo interno e limitare gli effetti della loro particolare disposizione nello spazio sulle quantità calcolate. La porosità esterna valutata all'interno del volume è un parametro indicativo in questo senso, poiché è fortemente influenzato dalla distribuzione dimensionale delle sfere e dalla geometria, e ha un effetto

determinante sulla fluidodinamica del sistema. Sono stati simulati nove casi a partire dalla stessa geometria di partenza (lo stesso file STL contenente le informazioni geometriche), estendendo progressivamente il volume incluso nel processo di costruzione della *mesh*. La geometria di partenza è stata generata nelle condizioni di massima deviazione standard della distribuzione dimensionale delle sfere, in modo tale che i risultati ottenuti fossero validi a maggior ragione anche nei casi con deviazione standard inferiore. In particolare, è stato impostato un diametro medio pari a  $90 \mu m$  e una deviazione standard di  $0.15 \times 10^6 \mu m^3$ . In Tab. 4.5 sono riportati le dimensioni del volume di controllo sperimentate nel corso dell’analisi. Per garantire condizioni omogenee nei diversi casi, è stata mantenuta una relazione lineare tra il volume considerato e il numero di celle generate. La porosità esterna calcolata per ogni caso considerato è riportata in Fig. 4.12. Come

Tabella 4.5: Volumi considerati nella selezione del REV. Le dimensioni del parallelepipedo sono espresse in multipli del diametro medio.

Caso	Dimensioni $x \times y \times z(\bar{D}_p)$
V1	$3 \times 3 \times 6$
V2	$4 \times 4 \times 8$
V3	$5 \times 5 \times 10$
V4	$5.5 \times 5.5 \times 11$
V5	$6 \times 6 \times 12$
V6	$6.5 \times 6.5 \times 13$
V7	$7 \times 7 \times 14$
V8	$7.5 \times 7.5 \times 15$
V9	$8 \times 8 \times 16$

si osserva dal grafico, al crescere del volume considerato il valore appare tendere ad un asintoto, collocato poco al di sotto del 39%. Si nota inoltre che la misura è affetta da rumore significativo, e non ha andamento monotono al crescere del volume. Per questo motivo, non è dimostrato un significativo miglioramento di accuratezza per volumi superiori al caso “V2”, e di conseguenza è possibile ritenere questo elemento di volume come rappresentativo delle condizioni globali del letto. La decisione è stata presa anche sulla base di considerazioni sul costo computazionale delle successive simulazioni fluidodinamiche. Come si osserva in Fig. 4.13, il numero di sfere contenute nel volume simulato per il caso “V2” (per la precisione, le sfere i quali centri sono collocati all’interno nel volume simulato) è compreso tra 150 e 200. In tutti i casi simulati per la costruzione del *dataset*, questo dato è compreso nello stesso intervallo.

#### 4.4.2 Discretizzazione spaziale e temporale

Allo scopo di verificare l’accuratezza della simulazione fluidodinamica, sono state effettuate quattro simulazioni utilizzando gli stessi parametri e la stessa geometria, variando la dimensione di cella della griglia base, ovvero sia la griglia strutturata a partire dalla

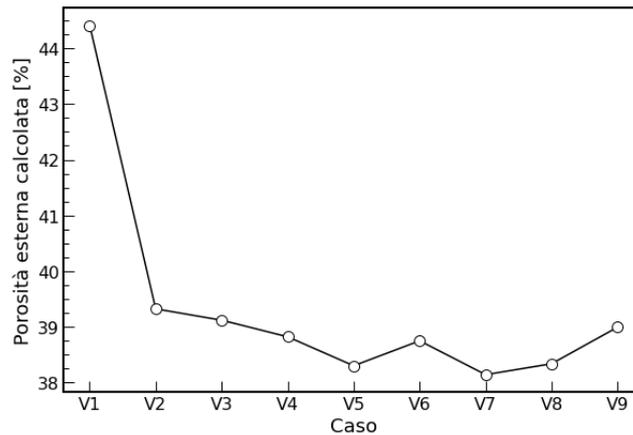


Figura 4.12: Porosità esterne calcolate nei diversi casi considerati in Tab. 4.5.

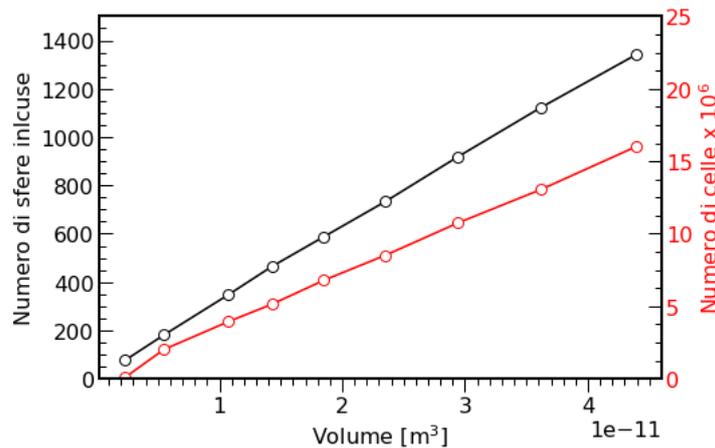


Figura 4.13: Numero di sfere incluse e numero di celle costruite all'interno del volume di simulazione.

quale si applica la procedura descritta alla sezione 2.2.3. Nel corso delle simulazioni effettuate, la griglia base è costituita da celle cubiche, di lato fissato pari ad una frazione del diametro medio delle sfere di resina. Le dimensioni di cella base sono riportate in Tab. 4.6. Come si osserva in Fig. 4.14, un grado di accuratezza sufficiente è ottenuto utilizzando una dimensione di cella base pari a  $1/24$  del diametro medio delle sfere. In Tab. 4.7 sono riassunti i parametri monitorati nel corso dell'analisi, in termini di qualità della griglia e di valore delle quantità calcolate. In Fig. 4.15 si riporta il particolare di una sezione della griglia di calcolo. A causa dei vincoli imposti sul numero massimo di celle costruite da `snappyHexMesh` all'interno del volume, pari a circa 10 milioni, si può notare come valori inferiori delle dimensioni di cella base permettano al *mesher* di generare raffinamenti locali più elevati in zone dove sono presenti superfici molto vicine tra

loro. Per quanto generalmente questo sia un aspetto positivo, il processo è difficilmente controllabile e introduce una certa disomogeneità nella griglia. Per questo motivo, si è preferito operare con mesh più omogenee e di qualità superiore. Inoltre, trattandosi di flusso in moto laminare, non si ha la formazione di un *boundary layer* sulla superficie delle particelle dovuto alla transizione delle condizioni del moto. Questo riduce la necessità di generare celle di dimensioni ridotte in prossimità della superficie del solido.

Tabella 4.6: Dimensioni di cella base considerate nel corso dell’analisi di *grid independence*, espresse in frazioni del diametro medio delle sfere.

Caso	Dimensioni $\times \bar{D}_p$
CD1	1/15
CD2	1/20
CD3	1/24
CD4	1/28

Tabella 4.7: Parametri monitorati nel corso dell’analisi di *grid independence*.

Caso	CD1	CD2	CD3	CD4
Celle totali $\times 10^6$	8.41	8.84	9.33	12.44
<i>aspect ratio</i> massimo	13.74	13.897	12.147	11.168
<i>non orthogonality</i> massima	56.551	56.835	55.996	54.134
<i>non orthogonality</i> media	12.760	12.708	12.288	11.490
<i>skewness</i> massima	3.694	3.795	3.661	3.348
Porosità esterna (%)	36.930	36.918	36.873	36.865
Area interfacciale specifica ( $10^4 m^{-1}$ )	2.459	2.449	2.437	2.433
Coeff. di Darcy ( $10^{11} m^2$ )	4.421	4.533	4.588	4.603
Coeff. di scambio termico ( $W/(m^2 K)$ )	68.226	69.778	67.687	67.827

La discretizzazione temporale utilizza un intervallo  $\Delta t$  fisso, pari a 0.0005 s. La scelta è stata condotta in modo tale da garantire che il numero di Courant massimo all’interno del dominio fluido fosse inferiore a 1, in modo tale da favorire la stabilità della simulazione. E stata altresì condotta un’analisi per verificare se a tale scelta corrispondesse il raggiungimento dell’indipendenza dei risultati dal valore di  $\Delta t$ . Come si osserva in Fig. 4.16, non è stata raggiunta una condizione di vera e propria indipendenza, ma si è ritenuto che l’accuratezza ottenuta imponendo  $\Delta t$  pari a 0.0005 s fosse sufficiente, anche sulla base di considerazioni sul costo computazionale delle simulazioni.

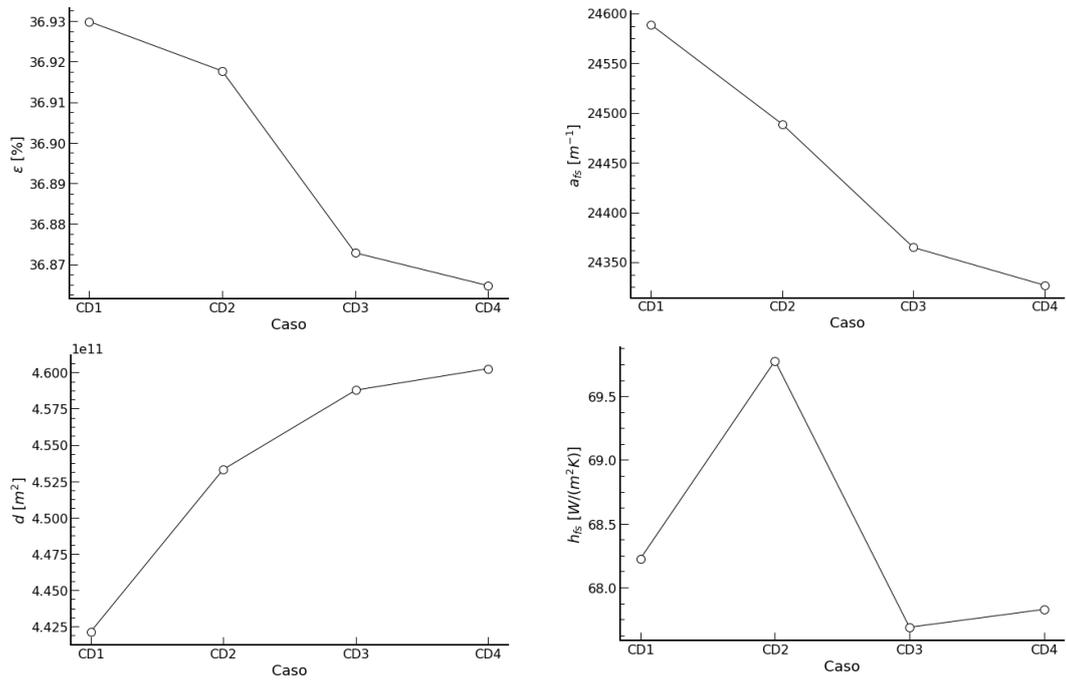


Figura 4.14: Valori calcolati di porosità esterna, area interfacciale specifica, coefficiente di Darcy e coefficiente di scambio termico a diverse dimensioni di cella base.

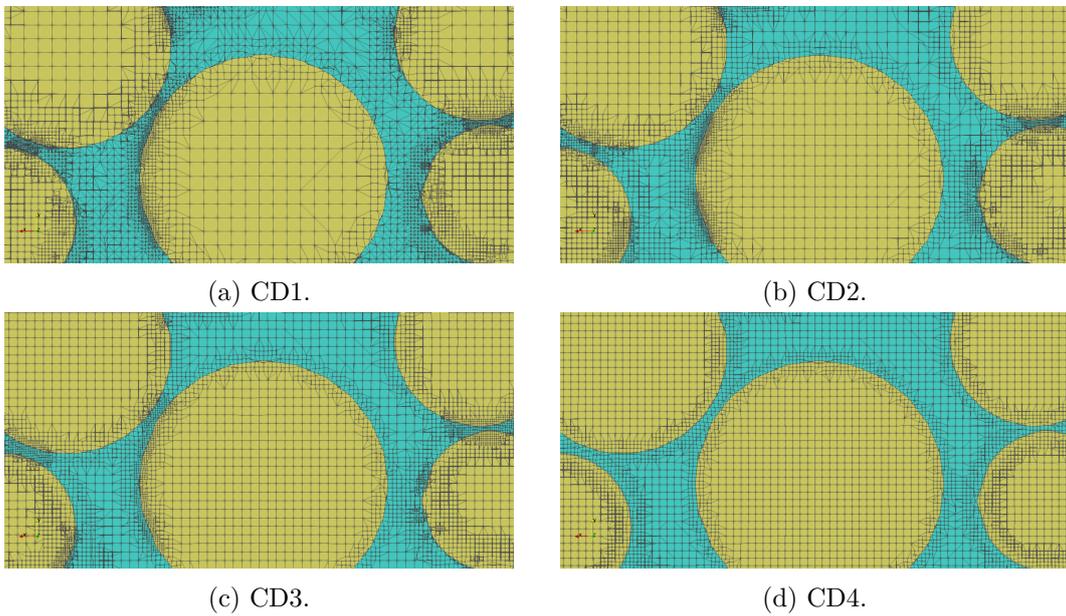


Figura 4.15: Particolare di una sezione della griglia di calcolo per diversi valori di cella base. Le dimensioni di cella sono riportate in Tab. 4.6.

Tabella 4.8: Discretizzazione temporale utilizzata nei diversi casi considerati nell’analisi di indipendenza.

Caso	Intervallo $\Delta t$ (s)
DT1	0.004
DT2	0.002
DT3	0.001
DT4	0.0005
DT5	0.00025

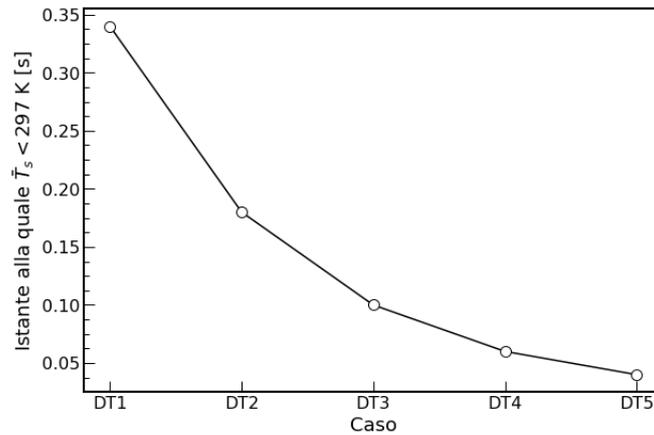


Figura 4.16: Istante alla quale la temperatura media nel dominio resina ( $\bar{T}_s$ ) è ridotta di un Kelvin rispetto alla temperatura iniziale.

#### 4.4.3 Analisi della varianza

Allo scopo di quantificare la varianza generata dalla processo stocastico di posizionamento iniziale delle sfere nella simulazione in **Blender** sulla geometria costruita e di conseguenza sulle quantità calcolate, sono state costruite diverse geometrie (generate da diverse *run* dello script in **blender**) utilizzando gli stessi parametri, scelti in modo arbitrario ad eccezione della deviazione standard, la quale è stata imposta pari al 95% del valore massimo allo scopo di ottenere dati che fossero rappresentativi anche dei casi con deviazione standard elevata, più problematici dal punto di vista della varianza. In particolare, si è utilizzato diametro medio pari a  $60 \mu m$  e deviazione standard pari a  $0.07 \mu m^3 \times 10^6$ . Le geometrie sono state successivamente inserite in cinque simulazioni fluidodinamiche aventi lo stesso gradiente di pressione e la stessa diffusività termica relativa della resina, pari rispettivamente a  $70000 Pa m$  e  $0.914$ ).

I risultati relativi al coefficiente di scambio termico necessitano di una trattazione a parte, sono quindi presentati nella sezione successiva.

Tabella 4.9: Quantità calcolate, valore medio e deviazione massima dal valor medio ottenuti nel corso dell'analisi della varianza.

Caso	Celle $\times 10^6$	$\epsilon$ (%)	$a_{fs}$ ( $10^4 m^{-1}$ )	$d$ ( $10^{11} m^2$ )
AV1	17.0	36.45	5.96	3.77
AV2	17.5	36.61	6.05	3.70
AV3	17.6	36.22	6.07	3.81
AV4	17.0	36.55	5.99	3.59
AV5	16.7	36.81	5.90	3.41
Media	-	36.53	5.99	3.66
Deviazione massima dalla media	-	0.847%	1.617%	6.711%

#### 4.4.4 Calcolo del coefficiente di scambio termico

Il modello oggetto del presente lavoro è stato concepito per produrre valori numerici delle quantità di interesse funzionali a una successiva implementazione degli stessi nel simulare i fenomeni di trasporto alla scala della colonna. In particolare, il coefficiente di scambio termico ottenuto attraverso la simulazione alla scala delle particelle può essere inserito all'interno di una simulazione alla scala della colonna utilizzando un modello di non-equilibrio termico locale in un mezzo poroso, attraverso un *solver* dedicato implementato in **OpenFOAM** dall'autore nel corso di un precedente lavoro di tirocinio presso GSK Vaccines. L'assunzione di non-equilibrio termico locale comporta la necessità di risolvere sullo stesso elemento di volume due equazioni accoppiate, relative al trasporto di calore nelle frazioni rispettivamente fluida e solida dell'elemento di volume. Le temperature del solido e del liquido sono definite come temperature medie delle rispettive frazioni all'interno del volume considerato (una singola cella della griglia computazionale). In particolare, le equazioni di trasporto dell'energia risolte nel modello di non-equilibrio macroscale sono [50]:

$$\epsilon \frac{\partial \bar{T}_f}{\partial t} = -\nabla \cdot (\mathbf{u} \bar{T}_f) + \alpha_f \nabla^2 \bar{T}_f + \frac{1}{\rho_f C_{p,f}} h_{fs} a_{fs} (\bar{T}_s - \bar{T}_f), \quad (4.18)$$

$$(1 - \epsilon) \frac{\partial \bar{T}_s}{\partial t} = \alpha_s \nabla^2 \bar{T}_s - \frac{1}{\rho_f C_{p,f}} h_{fs} a_{fs} (\bar{T}_s - \bar{T}_f). \quad (4.19)$$

Dove  $\bar{T}_{f,s}$  sono le temperature medie sui due domini,  $\alpha$  è la rispettiva diffusività termica,  $C_p$  il rispettivo calore specifico a pressione costante,  $a_{fs}$  e  $h_{fs}$  l'area interfacciale specifica e il coefficiente di scambio termico. Le due equazioni hanno in comune il termine relativo alla potenza termica scambiata (la sua densità, ovvero la potenza totale su volume) tra i due domini. Esso viene espresso come riportato nella seguente equazione:

$$|Q_{vol}| = |h_{fs} a_{fs} (\bar{T}_s - \bar{T}_f)|. \quad (4.20)$$

Da questa relazione, discende la definizione del coefficiente di scambio termico data dall'equazione:

$$h_{fs} = \left| \frac{Q_{vol}}{a_{fs}(\bar{T}_s - \bar{T}_f)} \right| = \left| \frac{q}{\bar{T}_s - \bar{T}_f} \right|. \quad (4.21)$$

Considerando il significato fisico di  $\bar{T}_f$  e  $\bar{T}_s$  nell'Eq. (4.21), nelle simulazioni alla scala delle particelle si è inizialmente voluto definire il coefficiente di scambio termico medio in maniera analoga. Di conseguenza, la differenza di temperatura di riferimento per il calcolo del coefficiente inizialmente utilizzata è stata  $\bar{T}_s - \bar{T}_f$ , dove in questo caso le due temperature rappresentano la temperatura nel rispettivo dominio mediata non sul volume della cella computazionale (come nel caso alla macroscale) ma sull'intero volume di simulazione. La scelta discende inoltre dal fatto che tale ragionamento possa essere applicabile anche nell'ambito del trasporto di materia.

Nel corso dell'analisi della varianza si è tuttavia osservato come il valore numerico del coefficiente così determinato sia soggetto ad una variabilità eccessiva. Il fenomeno è generato dal termine al denominatore dell'Eq. (4.21), poiché la differenza tra le temperature medie dei due domini è molto bassa e leggere oscillazioni di questa portano a scostamenti significativi del valore di  $h_{fs}$  calcolato. Questo fatto è dato dalla combinazione di due fattori:

- La bassa velocità del fluido all'interno del sistema;
- La vicinanza tra la diffusività termica del fluido e della resina.

All'interno del sistema infatti, il calore è trasportato prevalentemente per conduzione, e con velocità simile nei due domini. Di conseguenza, nel corso del transitorio si può osservare un fronte di temperatura omogeneo che si sposta dalla superficie di ingresso verso quella di uscita, come si riscontra in Fig. 4.17 e 4.18. Ciò giustifica le considerazioni fatte precedentemente. Il calcolo del coefficiente di scambio termico così definito non permette la costruzione di una rete neurale in grado di fare predizioni affidabili, perché il valore è scarsamente correlato con le *features* a causa della sua instabilità numerica. Per ovviare a questo problema, si è proseguito il lavoro utilizzando una differente definizione del coefficiente di scambio termico, data dalla seguente equazione:

$$h_{fs} = \left| \frac{q}{\bar{T}_f - T_{min}} \right|, \quad (4.22)$$

dove  $T_{min}$  è la temperatura minima sperimentata dal sistema, pari a 19.5 °C. In questo modo, non solo si riduce considerevolmente (un'ordine di grandezza) la varianza del coefficiente di scambio termico, ma si ottengono inoltre valori numerici compatibili con le correlazioni presenti in letteratura per lo scambio termico in regimi laminari a bassi numeri di Reynolds e Peclet ([51, 52]), in termini di ordine di grandezza. Le stesse fonti a cui si fa riferimento riportano la carenza di dati empirici affidabili per quanto riguarda il coefficiente di scambio termico in tali condizioni, a causa della limitata differenza di temperatura tra particelle solide e fluido circostante. In Tab. 4.10 si riporta il valore di varianza osservato per le due differenti definizioni di coefficiente di scambio termico utilizzate.

Ai fini del calcolo del coefficiente, sono stati considerati tutti i *timestep* per i quali la temperatura del media volumetrica del dominio solido è compresa tra i 24 e i 20.5 °C, escludendo quindi la fase iniziale e finale del transitorio. Il calcolo del coefficiente è stato effettuato ad ogni *timestep* selezionato, e successivamente ne è stata calcolata la media aritmetica.

Tabella 4.10: Deviazione massima dal valor medio del coefficiente di scambio termico osservata calcolandolo con diversi  $\Delta T$  di riferimento.

	$\Delta T_{ref} = \bar{T}_s - \bar{T}_f$	$\Delta T_{ref} = \bar{T}_f - T_{min}$
Deviazione massima dalla media	30.26%	3.87%

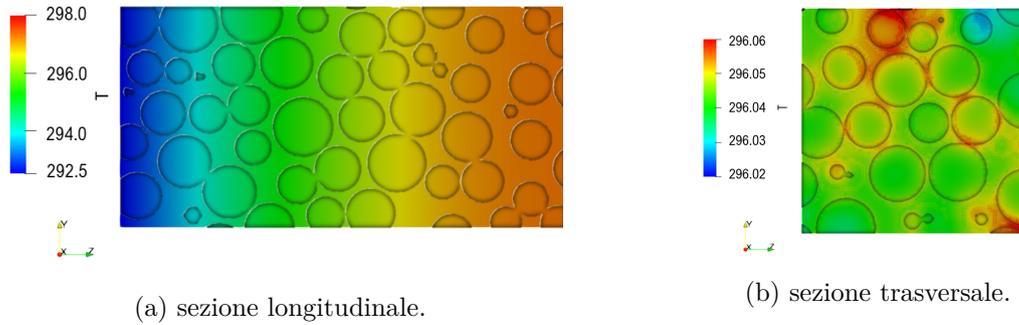


Figura 4.17: Profilo di temperatura lungo una sezione longitudinale e trasversale del dominio di simulazione a  $t = 1$  s. La temperatura è espressa in Kelvin.

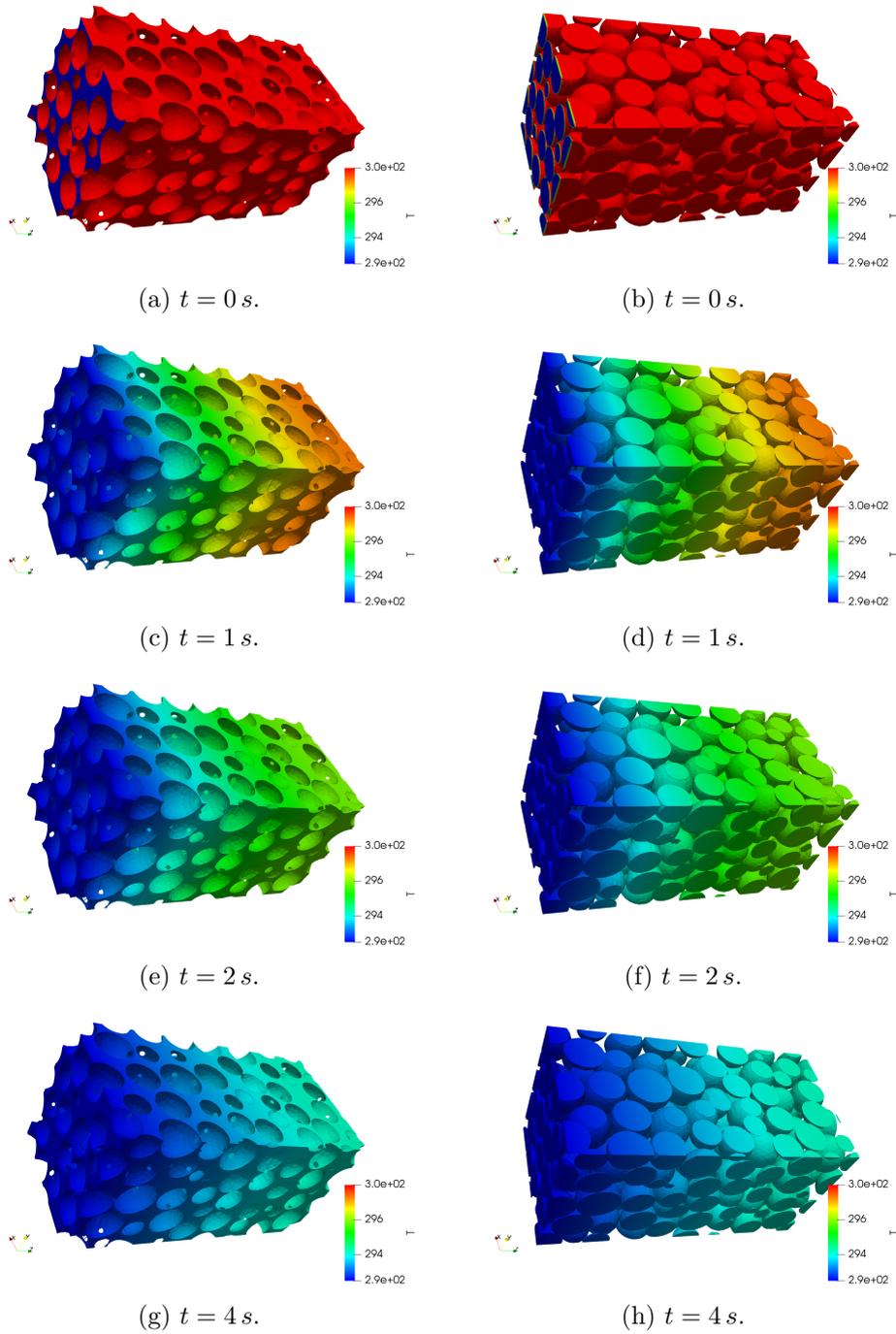


Figura 4.18: Esempio di evoluzione del profilo di temperatura nel dominio fluido (a sinistra) e nel dominio resina (a destra) a  $t = 0, 1, 2$  e  $4 s.$

## 4.5 Costruzione della rete neurale

Allo scopo di ottenere predizioni efficaci sulle quantità desiderate, sono state costruite due reti neurali indipendenti. La prima effettua predizioni sulle grandezze legate alla geometria del sistema, ovvero sia porosità esterna, area interfacciale specifica e coefficiente di Darcy, utilizzando come *features* il diametro medio e la deviazione standard della distribuzione volumetrica. La seconda predice il coefficiente di scambio termico medio utilizzando come *features* tutti i parametri considerati in tabella 4.4. Per la costruzione delle reti neurali, si è fatto uso della libreria *open-source TensorFlow* [53], attraverso l'interfaccia *Keras*, scritta in linguaggio *python*. Le reti create sono *shallow deep networks*, ovvero sia non includono connessioni ricorsive o all'indietro, e sono dotate di più *hidden layers*. Entrambe le reti utilizzano come algoritmo di ottimizzazione l'algoritmo "adam" [54], di cui si riporta brevemente il funzionamento. Esso è una modifica dell'algoritmo SGD, dotato di una buona robustezza numerica e in grado di fornire prestazioni generalmente più elevate ([41, 43]). Anch'esso valuta il gradiente locale nello spazio dei pesi della funzione di perdita, e aggiorna i pesi contro gradiente. L'elemento di novità introdotto rispetto all'algoritmo SGD è l'utilizzo di un *learning rate* adattivo, il quale assume un valore diverso per ogni parametro  $w_i$ . In particolare, l'algoritmo valuta la consistenza del gradiente nelle direzioni dei diversi parametri; se la sensitività della funzione di perdita è particolarmente elevata lungo alcuni assi (nello spazio dei pesi), risulta opportuno utilizzare un *learning rate* modesto per limitare l'instaurarsi di comportamenti oscillatori dell'algoritmo di ottimizzazione, viceversa in caso di direzioni caratterizzate da bassa sensitività. L'algoritmo prevede che il *learning rate* lungo una direzione venga scalato in funzione di grandezze correlate al valore assoluto della derivata parziale di  $f_{loss}$  in quella determinata direzione. Esso prevede l'utilizzo di quattro parametri definiti dall'utente, che sono *learning rate* ( $\lambda$ ), due parametri di decadimento ( $\rho_1$  e  $\rho_2$ , di valori compresi tra 0 e 1) e un parametro di condizionamento ( $\delta$ ). Nel corso del processo di aggiornamento computa due quantità: il valore mediato esponenzialmente di  $w_i$  ( $A_i$ , definito in Eq. (4.23)) e il valore mediato della derivata parziale della funzione di perdita ( $F_i$ , definito in Eq. (4.24)).

$$A_i \rightarrow \rho_1 A_i + (1 - \rho_1) \left( \frac{\partial f_{loss}}{\partial w_i} \right)^2, \quad (4.23)$$

$$F_i \rightarrow \rho_2 F_i + (1 - \rho_2) \left( \frac{\partial f_{loss}}{\partial w_i} \right). \quad (4.24)$$

Ad ogni iterazione, i pesi vengono aggiornati secondo l'Eq. (4.25), in cui  $\delta$  è un numero molto piccolo (in [54] si suggerisce di adottare  $10^{-8}$ ):

$$w_i \rightarrow w_i - \frac{\lambda_{adj}}{\sqrt{A_i} + \delta} F_i. \quad (4.25)$$

Poiché sia  $F_i$  che  $A_i$  sono inizializzati con valore nullo, le prime iterazioni sono affette da un significativo errore (*bias*, ovvero sia l'errore non correlato alla rumorosità dei dati). Per limitare il fenomeno, si introduce un fattore correttivo funzione del numero di iterazioni eseguite ( $e$ ), il quale tende a 1 al crescere di  $e$ . Di conseguenza il termine  $\lambda_{adj}$ , ovvero sia il

*learning rate* alla quale è stata applicata la correzione, viene calcolato come nella seguente equazione:

$$\lambda_{adj} = \lambda \left( \frac{\sqrt{1 - \rho_1^\epsilon}}{1 - \rho_2^\epsilon} \right). \quad (4.26)$$

Poiché sia *features* che *labels* assumono valori su scale diverse, a diversi ordini di grandezza, si rende necessario operare un *rescaling* prima di allenare e di utilizzare il modello. Questa operazione permette di riformulare il *dataset* in modo tale da avere parametri che variano all'interno dello stesso intervallo di valori, comportando nella maggior parte dei casi una migliore prestazione predittiva del modello grazie al miglior condizionamento dell'algoritmo di aggiornamento dei parametri [41]. Nel presente modello si è utilizzata la standardizzazione di *features* e *labels*, che riconduce i rispettivi valori ad una distribuzione normale, centrata in zero e avente deviazione standard unitaria, secondo l'equazione:

$$\hat{x}_{k,j} = \frac{x_{k,j} - \text{mean}(\mathbf{x}_k)}{\text{std}(\mathbf{x}_k)}, \quad (4.27)$$

dove  $\hat{x}_{k,j}$  è il valore standardizzato dell'elemento di un *dataset* composto da  $i$  *features/labels* e  $j$  *samples*.



## 5. Discussione dei risultati

In questo capitolo vengono presentate e analizzate le prestazioni delle reti neurali costruite. Il processo di allenamento delle reti è stato monitorato attraverso un insieme di 15 *samples* (*validation set*), non inclusi nel processo di allenamento, per verificare l'andamento dell'errore di generalizzazione e individuare un eventuale *underfitting/overfitting* dei dati. Successivamente, le reti sono state valutate comparando le predizioni effettuate sui 15 *samples* del *test set*, anch'essi non inclusi nell'allenamento, con i rispettivi valori calcolati nelle simulazioni CFD effettuate. Sia *validation* che *test set* sono stati mantenuti fissi nell'allenamento di entrambe le reti. Sono state sperimentate diverse combinazioni di iperparametri per valutarne l'effetto sulle prestazioni e per individuare una condizione di ottimo. In Fig. 5.1 sono riportati degli esempi dell'andamento della funzione di perdita (l'errore assoluto medio, MAE) su *training* e *validation set* in caso di *underfitting* e *overfitting*. Il primo caso si verifica quando le iterazioni (o *epochs*) eseguite non sono sufficienti a individuare un punto di minimo di  $f_{loss}$ , di conseguenza sia la curva degli errori sul *training set* che quella sul *validation set* non mostrano il raggiungimento di un valore asintotico, che corrisponde ad un punto di minimo locale di  $f_{loss}$ . Il secondo caso si verifica quando il modello possiede capacità elevata ed è allenato sufficientemente a lungo da perdere di generalità, poiché adattandosi eccessivamente ai *labels* del *training set*, è fortemente influenzato dal rumore che caratterizza questi ultimi; questo si traduce in una crescita dell'errore sul *validation set*.

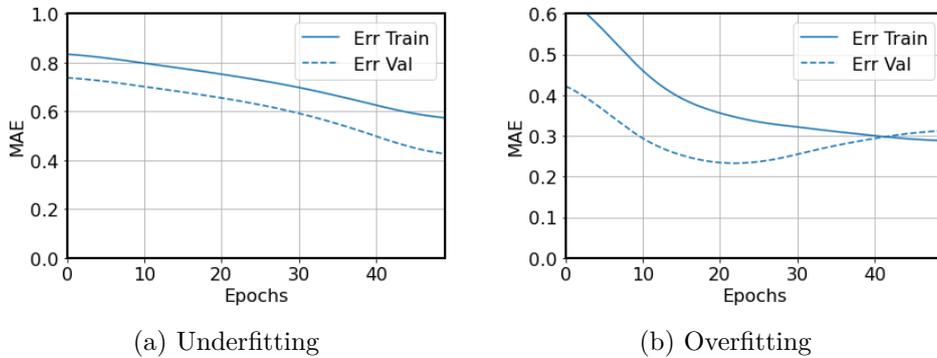


Figura 5.1: Esempio dell'andamento della funzione di perdita, l'errore assoluto medio, su *training* e *validation set* in caso di *underfitting* e *overfitting* dei dati.

### 5.1 Rete predittiva dei parametri geometrici

Una prima rete neurale è stata costruita allo scopo di eseguire predizioni sulla porosità esterna, l'area interfacciale specifica e il coefficiente di Darcy noto il diametro medio delle *beads* e la deviazione standard della distribuzione volumetrica dei diametri. La rete è stata

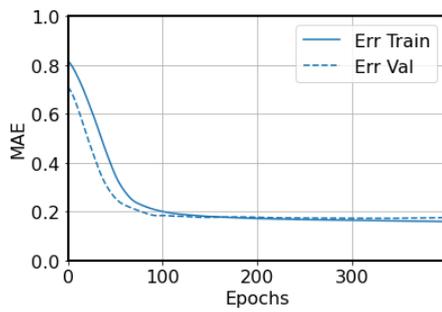
valutata attraverso gli errori relativi delle predizioni effettuate sui *samples* del *test set*. In particolare, la condizione di ottimo è stata individuata allenando la rete cinque volte per ogni combinazione di iperparametri sperimentata e calcolando la media aritmetica dell'errore relativo medio e massimo. Poiché gli insiemi sono mantenuti fissi nel corso dell'allenamento, le discrepanze tra i risultati ottenuti a parità di iperparametri utilizzati sono da attribuirsi all'inizializzazione casuale dei pesi e alla conseguente individuazione di differenti punti di minimo locale di  $f_{loss}$ . Il calcolo del gradiente  $\nabla_w f_{loss}$ , necessario per l'aggiornamento dei pesi, è stato effettuato sull'intero *training set*. In Tab. 5.1 sono riportate le combinazioni di iperparametri sperimentate e le rispettive medie degli errori relativi medi sulle quantità predette. Nel corso dell'indagine, sono stati variati il numero di *layers*, il numero di neuroni per *layer* e il *learning rate* ( $\lambda$ ), mentre i parametri di decadimento e il parametro di condizionamento sono stati mantenuti fissi al valore suggerito in letteratura [54], ovvero  $\rho_1 = 0.9$ ,  $\rho_2 = 0.999$ ,  $\delta = 10^{-8}$ . Il numero di *epochs* è stato selezionato monitorando l'andamento di  $f_{loss}$  nel corso dell'allenamento. La funzione di attivazione utilizzata per tutti i neuroni della rete è la funzione ReLU (*Rectified Linear Unit*).

Sulla base dei dati riportati in tabella, la combinazione ottimale selezionata corrisponde a un *learning rate* pari a 0.0005 e a un'architettura composta da quattro *hidden layers* da 64 neuroni ciascuno. La rete così costruita produce errori relativi massimi su porosità esterna, area interfacciale specifica e coefficiente di Darcy rispettivamente pari a 2.805%, 2.715% e 15.60%. In Fig. 5.2 è riportato l'andamento dell'errore assoluto medio e dello scarto quadratico medio sugli insiemi di *training* e *validation* nel corso dell'allenamento. In Fig. 5.3 e sono riportati i diagrammi di parità relativi alle predizioni effettuate sul *test set*, mentre in Fig. 5.4 è rappresentata la distribuzione degli errori assoluti e relativi, suddivisi per le varie quantità. Dalla Tab. 5.1 si nota come il valore degli errori non si modifichi significativamente al variare dei parametri, di conseguenza è ragionevole affermare che il limite principale all'accuratezza della rete è rappresentato dalla limitatezza del *dataset* e dalla qualità dei dati raccolti. Allo scopo di verificare l'affermazione precedente, si è allenata la stessa rete utilizzando solo una parte del *training set* per valutarne l'impatto sulle prestazioni. In particolare, l'allenamento è stato eseguito cinque volte su *samples* estratti casualmente dal *training set* pari al 50% del totale; la procedura è stata in seguito ripetuta per un campione pari al 75% del totale. Come si osserva da Tab. 5.2, la dimensione del *dataset* ha effetti significativi solamente sulla predizione del coefficiente di Darcy e, in misura minore, sull'area interfacciale specifica; di conseguenza è ragionevole supporre che la raccolta di ulteriori dati con la stessa metodologia possa avere un effetto positivo esclusivamente sulla predizione di queste quantità.

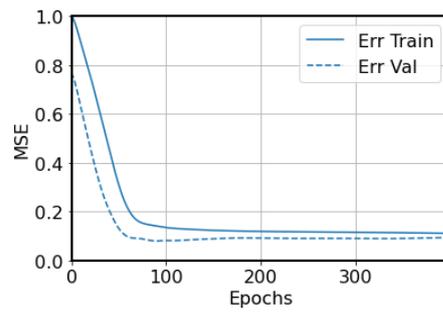
A scopo di confronto, in Tab. 5.3 si riportano la varianza calcolata delle grandezze in questione nel corso dell'analisi riportata in sezione 4.4.3 e il rispettivo errore relativo medio sulle predizioni. Come si osserva dalla tabella, i valori di errore relativo medio non si discostano molto dai valori di varianza calcolata, indice del corretto andamento dell'allenamento della rete e della sua sufficiente capacità predittiva. Per due delle tre quantità calcolate, l'errore relativo medio è inferiore del rispettivo dato sulla varianza.

Tabella 5.1: Combinazioni di iperparametri testate e rispettivi errori relativi medi sulle quantità predette. Il numero di elementi in parentesi quadre rappresenta il numero di *hidden layer*, mentre il valore rappresenta il numero di neuroni per ciascun *layer*.

Architettura	ER $\epsilon$ (%)	ER $a_{fs}$ (%)	ER $d$ (%)
$\lambda = 0.0005$			
[64 64]	1.149	1.600	8.208
[32 64 32]	1.154	1.227	6.699
[64 64 64]	1.131	1.047	6.043
[32 64 64 32]	1.123	0.988	5.428
<b>[64 64 64 64]</b>	<b>1.106</b>	<b>0.929</b>	<b>5.149</b>
[64 64 64 64 64]	1.082	1.430	7.645
$\lambda = 0.001$			
[64 64]	1.129	1.025	6.187
[32 64 32]	1.131	1.371	7.523
[64 64 64]	1.128	1.133	6.360
[32 64 64 32]	1.120	1.134	6.312
[64 64 64 64]	1.104	1.278	7.075
[64 64 64 64 64]	1.069	1.539	7.636
$\lambda = 0.005$			
[64 64]	1.158	1.059	6.424
[32 64 32]	1.128	1.231	6.147
[64 64 64]	1.155	1.334	5.828
[32 64 64 32]	1.065	1.587	7.289
[64 64 64 64]	1.113	1.433	6.561
[64 64 64 64 64]	1.068	2.287	9.304



(a) MAE



(b) MSE

Figura 5.2: Andamento dell'errore assoluto medio (MAE) e dello scarto quadratico medio (MSE) nel corso dell'allenamento della rete per la predizione dei parametri geometrici.

Tabella 5.2: Errori relativi sulle predizioni di porosità esterna ( $\epsilon$ ), area interfacciale specifica ( $a_{fs}$ ) e coefficiente di Darcy ( $d$ ) al variare della dimensione del *dataset* di allenamento.

<b><i>Samples</i> di allenamento</b>	<b>ER medio <math>\epsilon</math> (%)</b>	<b>ER medio <math>a_{fs}</math> (%)</b>	<b>ER medio <math>d</math> (%)</b>
<b>50</b>	1.135	1.460	8.165
<b>75</b>	1.167	1.213	7.203
<b>100</b>	1.106	0.929	5.149

Tabella 5.3: Varianza calcolata e errore relativo medio sulle predizioni di porosità esterna, area interfacciale specifica e coefficiente di Darcy.

	<b>Varianza calcolata (%)</b>	<b>ER medio sulle predizioni (%)</b>
$\epsilon$	0.847	1.106
$a_{fs}$	1.617	0.929
$d$	6.711	5.149

## 5.2 Rete predittiva del coefficiente di scambio termico

Una seconda rete neurale è stata costruita allo scopo di predire il coefficiente di scambio termico tra fluido e resina, noti il diametro medio delle *beads*, la deviazione standard della distribuzione volumetrica dei diametri, la diffusività termica della resina, espressa in rapporto a quella della fase fluida, e il numero di Reynolds riferito alla particella. La rete è stata valutata attraverso l'errore assoluto e relativo sulle predizioni effettuate sui *samples* del *test set*. Anche in questo caso, la condizione di ottimo è stata determinata allenando cinque volte la rete per ogni combinazione di parametri sperimentata. Il calcolo del gradiente di  $f_{loss}$  in questo caso non è stato effettuato sull'intero *training set* in un singolo passaggio, ma su sottoinsiemi di questo costituite da 25 *samples* (i cosiddetti *minibatches*). La scelta è stata intrapresa poiché si è verificato che questo accorgimento comporta un discreto miglioramento delle prestazioni della rete. La proprietà regolarizzante dell'uso dei *minibatches* nei confronti del rumore di cui sono affetti i *labels* del *training set* è nota in letteratura, e permette di ridurre l'errore di generalizzazione della rete [55]. Le combinazioni di iperparametri sperimentate hanno seguito i criteri esposti alla sezione 5.1. In Tab. 5.4 sono riportati i casi considerati e gli errori assoluti e relativi medi osservati.

La combinazione di iperparametri ottimale individuata corrisponde a un *learning rate* pari a 0.005 e a un'architettura composta da tre *hidden layers*, aventi rispettivamente 10, 10, e 5 neuroni. La rete così costruita produce un errore relativo massimo, mediato su cinque allenamenti, pari al 15.06%. In Fig 5.5 si riporta l'andamento dell'errore assoluto

Tabella 5.4: Combinazioni di iperparametri testate e rispettivi errori assoluti e relativi medi sulle predizioni. Il numero di elementi in parentesi quadre rappresenta il numero di *hidden layer*, mentre il valore rappresenta il numero di neuroni per ciascun *layer*.

Architettura	EA $h_{fs}$ ( $W/(m^2K)$ )	ER $h_{fs}$ (%)
$\lambda = 0.0005$		
[50]	3.084	7.625
[100]	3.165	8.041
[10 10]	3.389	8.447
[15 15]	3.142	7.789
[20 20]	3.036	7.444
[10 10 5]	3.212	8.096
[10 10 10]	3.494	8.747
[15 10 5]	3.367	8.472
$\lambda = 0.001$		
[50]	3.164	8.048
[100]	3.085	7.884
[10 10]	3.398	8.445
[15 15]	3.282	8.392
[20 20]	3.076	7.711
[10 10 5]	3.470	8.716
[10 10 10]	3.183	7.642
[15 10 5]	3.128	7.829
$\lambda = 0.005$		
[50]	3.122	8.024
[100]	2.994	7.702
[10 10]	3.355	8.548
[15 15]	3.110	8.006
[20 20]	3.205	8.094
<b>[10 10 5]</b>	<b>2.802</b>	<b>7.004</b>
[10 10 10]	3.359	8.633
[15 10 5]	3.228	8.152

medio e dello scarto quadratico medio sugli insiemi di *training* e *validation* nel corso dell'allenamento. Dal grafico si osserva come il valore di  $f_{loss}$  rimanga elevato per entrambi gli insiemi; in particolare, considerato il valore significativo dell'errore sul *training set*, è ragionevole presumere che la capacità predittiva del modello sia limitata dalla qualità dei dati raccolti, come suggerito in [43]. L'elevato rumore che caratterizza i valori del coefficiente, legato alle condizioni di simulazione relative al trasporto di calore nel sistema e alla metodologia di calcolo, come esposto in sezione 4.4.4, rappresenta una difficoltà intrinseca per la costruzione di un modello predittivo. Per questo motivo, le prestazioni

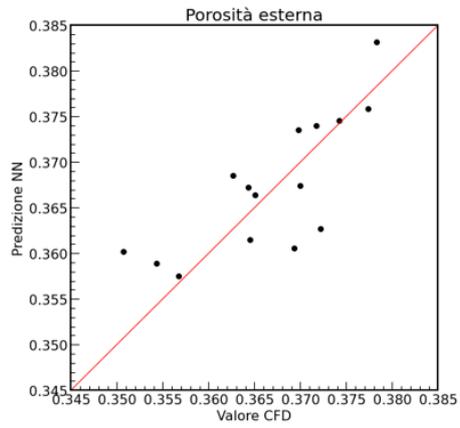
della rete, presentate in Fig. 5.6 e 5.7, sono da ritenersi solo parzialmente soddisfacenti; si rende quindi necessario individuare una procedura alternativa per il calcolo del coefficiente di scambio termico. Per verificare l'effetto sulle predizioni della dimensione del *dataset* di allenamento, la rete è stata allenata rispettivamente cinque volte su frazioni casuali del *training set*, corrispondenti al 50% e 75% del totale dei *samples*. Come si osserva da Tab. 5.5, al crescere del numero di *samples* utilizzati per l'allenamento si ha un discreto miglioramento nell'accuratezza delle predizioni. In Tab. 5.6 si riportano la varianza calcolata e l'errore relativo medio sulle predizioni. In questo caso, l'errore ottenuto dalla rete è significativamente più elevato della varianza. Poiché quest'ultima è stata calcolata utilizzando un numero limitato di simulazioni, è possibile che sia stata sottostimata e ciò giustifica la discrepanza tra i due valori.

Tabella 5.5: Errori relativi sulle predizioni della rete al variare della dimensione del *dataset* di allenamento.

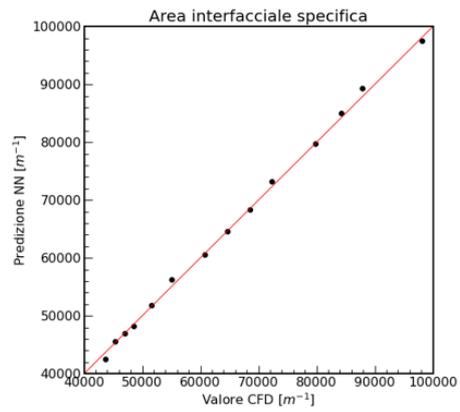
<b><i>Samples</i> di allenamento</b>	<b>EA medio <math>h_{fs}</math> (<math>W/(m^2K)</math>)</b>	<b>ER medio <math>h_{fs}</math> (%)</b>
<b>50</b>	3.531	8.932
<b>75</b>	3.376	8.535
<b>100</b>	2.802	7.004

Tabella 5.6: Varianza calcolata e errore relativo medio sulle predizioni del coefficiente di scambio termico.

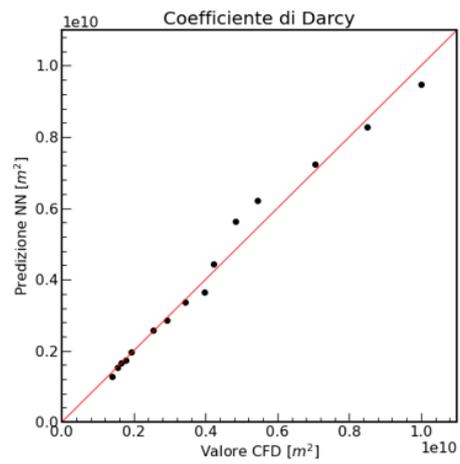
	<b>Varianza calcolata (%)</b>	<b>ER medio sulle predizioni (%)</b>
<b><math>h_{fs}</math></b>	3.873	7.004



(a) Porosità esterna



(b) Area interfacciale specifica



(c) Coeff. di Darcy

Figura 5.3: Diagrammi di parità delle predizioni effettuate sul *test set*.

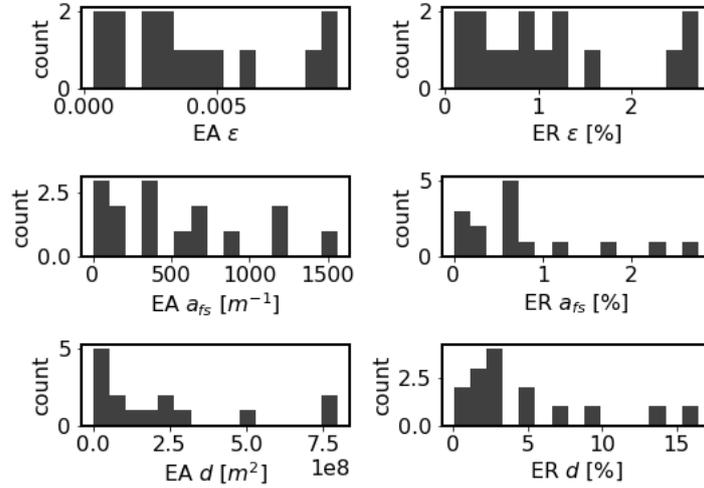


Figura 5.4: Distribuzioni degli errori assoluti e relativi sulle predizioni effettuate dalla rete per la predizione di porosità esterna ( $\epsilon$ ), area interfacciale specifica ( $a_{fs}$ ) e coefficiente di Darcy ( $d$ ).

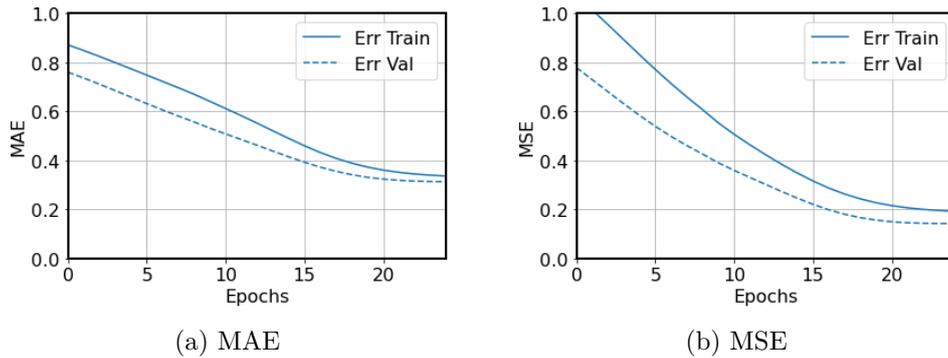


Figura 5.5: Andamento dell'errore assoluto medio (MAE) e dello scarto quadratico medio (MSE) nel corso dell'allenamento della rete per la predizione del coefficiente di scambio termico.

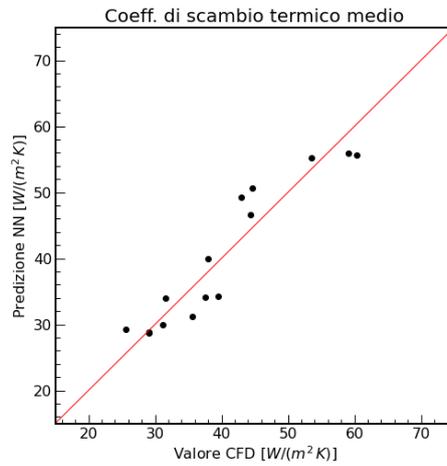


Figura 5.6: Diagramma di parità delle predizioni del coefficiente di scambio termico effettuate sul *test set*.

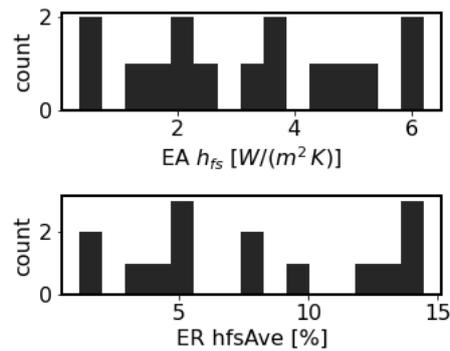


Figura 5.7: Distribuzioni degli errori assoluti e relativi sulle predizioni effettuate dalla rete per la predizione del coefficiente di scambio termico.



## 6. Conclusioni e prospettive future

L'obiettivo del presente lavoro è stato proporre una metodologia innovativa, applicata prevalentemente in contesti accademici, per l'ottenimento di informazioni di interesse industriale nell'ambito di processi di separazione cromatografica. L'elemento di novità introdotto consiste nell'ottenere attraverso simulazioni di fluidodinamica computazionale parametri di interesse relativi da un lato alle proprietà geometriche di letti impaccati non strutturati di colonne cromatografiche, dall'altro alle proprietà di trasporto di calore all'interno del letto e in particolare allo scambio termico tra il fluido e le particelle di resina. Il lavoro è stato strutturato in modo tale da essere integrabile con un modello CFD del trasporto di calore nell'intera colonna cromatografica facente uso di un *solver* che implementa la condizione di non-equilibrio termico locale, sviluppato dall'autore nel corso di un precedente lavoro di tirocinio. Inoltre, le condizioni al contorno e i metodi di calcolo delle quantità di interesse sono stati selezionati per essere estendibili, con modifiche minori, a problemi di scambio di materia all'interno del letto in colonne cromatografiche a esclusione dimensionale o di altro tipo, in quest'ultimo caso previa implementazione di un modello di interazione soluto-resina. I dati raccolti nel corso delle simulazioni sono stati in seguito utilizzati per la costruzione di un modello predittivo basato su reti neurali, allo scopo di valutarne le prestazioni ottenibili a partire da un *dataset* limitato, individuarne le criticità e determinare le potenzialità future di questa metodologia, a fronte dell'importante risparmio di tempo, risorse e costo computazionale che essa offre.

Allo scopo di costruire un *dataset* idoneo alla costruzione del modello predittivo, è stato sviluppato un *workflow* sistematico che permette di ottenere per via numerica la porosità esterna, l'area interfacciale specifica, coefficiente di Darcy e coefficiente di scambio termico relativi a letti cromatografici costituiti da particelle sferiche polidisperse di resina, impaccate in modo casuale. Allo scopo di predire la porosità esterna, l'area interfacciale specifica e il coefficiente di Darcy sono state individuate come variabili causali il diametro medio delle particelle di resina e la deviazione standard della distribuzione volumetrica dei diametri; mentre per la predizione del coefficiente sono stati utilizzati il diametro medio, la deviazione standard sulla distribuzione volumetrica dei diametri, la diffusività termica delle particelle e il numero di Reynolds (riferito alla particella) del sistema. La procedura di simulazione prevede:

- la costruzione di una geometria rappresentativa della struttura del *bulk* del letto impaccato;
- l'elaborazione della geometria creata attraverso l'implementazione del *bridges method* per la modifica locale dei punti di contatto tra le sfere di resina;
- l'esecuzione delle simulazioni CFD necessarie all'ottenimento delle quantità di interesse;
- l'estrazione e il *postprocessing* dei dati provenienti dalle simulazioni.

In seguito, sono stati identificati gli intervalli di indagine delle variabili causali, in modo tale che fossero aderenti con le usuali condizioni operative di colonne cromatografiche deputate alla purificazione di prodotti proteici. In questo modo, si è proceduto alla costruzione di un *dataset* generato da combinazioni casuali delle variabili all'interno dei rispettivi intervalli, poi inserite all'interno delle simulazioni. Nel corso del lavoro si è fatto uso esclusivamente di strumenti informatici *open source*.

Per quanto riguarda la predizione di porosità esterna, area interfacciale specifica e coefficiente di Darcy, la rete neurale costruita produce errori relativi medi sulle quantità calcolate rispettivamente pari a 1.106%, 0.929% e 5.149%. Si rileva come questi valori siano di entità comparabile con la varianza calcolata per le rispettive quantità, dovuta all'aleatorietà del processo di creazione della geometria e alla dimensione del volume di simulazione, limitata a causa del costo computazionale che comporta. In particolare, a seguito delle indagini condotte è ragionevole ritenere che gli errori della rete siano legati alla limitata accuratezza del modello CFD e, nel caso della predizione del coefficiente di Darcy, anche alla scarsità di dati raccolti. Si conclude che l'eventuale perfezionamento del modello necessita di una modifica della procedura di simulazione fluidodinamica, in particolare considerando un volume di dimensioni maggiori per ridurre la varianza sulle quantità calcolate.

Nel caso della predizione del coefficiente di scambio termico, la rete neurale costruita produce un errore relativo medio pari a 7.004%, dato significativamente più elevato rispetto alla varianza calcolata della quantità in esame. Si ritiene che il fenomeno sia dovuto sia alla possibile sottostima della varianza sia al ridotto numero di *samples* considerati. Di conseguenza, anche in questo caso l'eventuale perfezionamento della rete passa attraverso una modifica della procedura di simulazione fluidodinamica, e in particolare dall'utilizzo di condizioni al contorno differenti per limitare le criticità relative al calcolo del coefficiente, legate alla ridotta differenza di temperatura tra il fluido e la resina all'interno del dominio simulato.

I risultati ottenuti dimostrano la possibilità di costruire modelli predittivi basati su *shallow deep neural networks*, caratterizzati da una sufficiente accuratezza e da un ridottissimo costo computazionale, e delineano alcune soluzioni per migliorarne le prestazioni.

Si vuole infine mettere in luce la necessità di convalidare e ottimizzare il *workflow* di simulazione proposto. Per poter applicare con successo la procedura qui delineata in applicazioni reali, è necessaria una conferma dell'accuratezza e della precisione relative al calcolo delle quantità di interesse, non ottenuta nel corso del presente studio per l'indisponibilità di dati sperimentali di riferimento.

## Bibliografia

- [1] M. Grieves, “Digital twin: manufacturing excellence through virtual factory replication”, *White paper*, vol. 1, pp. 1–7, 2014.
- [2] E. Glaessgen and D. Stargel, “The digital twin paradigm for future nasa and us air force vehicles”, in *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, p. 1818, 2012.
- [3] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, “Digital twin in industry: State-of-the-art”, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.
- [4] J. Smith, M. Lipsitch, and J. W. Almond, “Vaccine production, distribution, access, and uptake”, *The Lancet*, vol. 378, no. 9789, pp. 428–438, 2011.
- [5] J. Ehreth, “The global value of vaccination”, *Vaccine*, vol. 21, no. 7-8, pp. 596–600, 2003.
- [6] G. Walsh, *Biopharmaceuticals: Biochemistry and Biotechnology*. Wiley, 2006.
- [7] R. J. Silva, J. P. Mota, C. Peixoto, P. M. Alves, and M. J. Carrondo, “Improving the downstream processing of vaccine and gene therapy vectors with continuous chromatography”, *Pharmaceutical Bioprocessing*, vol. 3, no. 8, pp. 489–505, 2015.
- [8] R. Morenweiser, “Downstream processing of viral vectors and vaccines”, *Gene therapy*, vol. 12, no. 1, pp. S103–S110, 2005.
- [9] J. Bailey and D. Ollis, *Biochemical Engineering Fundamentals*. McGraw-Hill chemical engineering series, McGraw-Hill, 1986.
- [10] N. Pathak, C. Norman, S. Kundu, S. Nulu, M. Durst, and Z. Fang, “Modeling flow distribution in large-scale chromatographic columns with computational fluid dynamics”, *BioProcess International*, vol. 6, pp. 72–81, 2008.
- [11] G. Jagschies, E. Lindskog, K. Lacki, and P. Galliher, *Biopharmaceutical Processing: Development, Design, and Implementation of Manufacturing Processes*. Elsevier Science, 2018.
- [12] A. S. Rathore, D. Kumar, and N. Kateja, “Recent developments in chromatographic purification of biopharmaceuticals”, *Biotechnology letters*, vol. 40, no. 6, pp. 895–905, 2018.
- [13] M. Zhao, M. Vandersluis, J. Stout, U. Haupts, M. Sanders, and R. Jacquemart, “Affinity chromatography for vaccines manufacturing: Finally ready for prime time?”, *Vaccine*, vol. 37, no. 36, pp. 5491–5503, 2019.
- [14] G. Najafpour, *Biochemical engineering and biotechnology*. Elsevier, 2015.
- [15] M. L. Shuler and F. Kargi, *Bioprocess Engineering: Basic Concepts*. Prentice Hall, 2nd ed., 2006.
- [16] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, and B. Van Wachem, *Computational fluid dynamics for engineers*. Cambridge university press, 2011.

- 
- [17] C. Greenshields, “Openfoam v8 user guide: 4.5 numerical schemes”. <https://cfd.direct/openfoam/user-guide/v8-fvSchemes>, Jul 2020.
- [18] C. Greenshields, “Openfoam v8 user guide: 5.4 mesh generation with snappyhexmesh”. <https://cfd.direct/openfoam/user-guide/v8-snappyHexMesh>, Jul 2020.
- [19] A. T. Hanke and M. Ottens, “Purifying biopharmaceuticals: knowledge-based chromatographic process development”, *Trends in biotechnology*, vol. 32, no. 4, pp. 210–220, 2014.
- [20] V. Kumar and A. M. Lenhoff, “Mechanistic modeling of preparative column chromatography for biotherapeutics”, *Annual Review of Chemical and Biomolecular Engineering*, vol. 11, pp. 235–255, 2020.
- [21] A. G. Dixon and B. Partopour, “Computational fluid dynamics for fixed bed reactor design”, *Annual Review of Chemical and Biomolecular Engineering*, vol. 11, pp. 109–130, 2020.
- [22] J. Chou, A. Lennart, J. Wong, M. F. Ali, P. N. Floriano, N. Christodoulides, J. Camp, and J. T. McDevitt, “Modeling analyte transport and capture in porous bead sensors”, *Analytical chemistry*, vol. 84, no. 5, pp. 2569–2575, 2012.
- [23] T. D. Scheibe, W. A. Perkins, M. C. Richmond, M. I. McKinley, P. D. Romero-Gomez, M. Ostrom, T. W. Wietsma, J. A. Serkowski, and J. M. Zachara, “Pore-scale and multiscale numerical simulation of flow and transport in a laboratory-scale column”, *Water Resources Research*, vol. 51, no. 2, pp. 1023–1035, 2015.
- [24] M. Giese, K. Rottschafer, and D. Vortmeyer, “Measured and modeled superficial flow profiles in packed beds with liquid flow”, *American Institute of Chemical Engineers. AIChE Journal*, vol. 44, no. 2, p. 484, 1998.
- [25] L. Leal, “4 challenges and opportunities in fluid mechanics and transport phenomena”, in *Advances in chemical engineering*, vol. 16, pp. 61–79, Elsevier, 1991.
- [26] A. G. Dixon, M. Nijemeisland, and H. Stitt, “Cfd simulation of reaction and heat transfer near the wall of a fixed bed”, *International Journal of Chemical Reactor Engineering*, vol. 1, no. 1, 2003.
- [27] S. Khirevich, A. Hölzel, A. Seidel-Morgenstern, and U. Tallarek, “Time and length scales of eddy dispersion in chromatographic beds”, *Analytical chemistry*, vol. 81, no. 16, pp. 7057–7066, 2009.
- [28] E. Izadi and A. Bezuijen, “Simulating direct shear tests with the bullet physics library: A validation study”, *PLOS one*, vol. 13, no. 4, p. e0195073, 2018.
- [29] G. Boccardo, F. Augier, Y. Haroun, D. Ferre, and D. L. Marchisio, “Validation of a novel open-source work-flow for the simulation of packed-bed reactors”, *Chemical Engineering Journal*, vol. 279, pp. 809–820, 2015.
- [30] B. Partopour and A. G. Dixon, “An integrated workflow for resolved-particle packed bed models with complex particle shapes”, *Powder Technology*, vol. 322, pp. 258–272, 2017.
- [31] M. Dorn and D. Hekmat, “Simulation of the dynamic packing behavior of preparative chromatography columns via discrete particle modeling”, *Biotechnology Progress*, vol. 32, no. 2, pp. 363–371, 2016.

- 
- [32] M. Dalman, J. Merkin, and C. McGreavy, “Fluid flow and heat transfer past two spheres in a cylindrical tube”, *Computers & fluids*, vol. 14, no. 3, pp. 267–281, 1986.
- [33] B. Lloyd and R. Boehm, “Flow and heat transfer around a linear array of spheres”, *Numerical Heat Transfer, Part A Applications*, vol. 26, no. 2, pp. 237–252, 1994.
- [34] O. R. Derkx and A. G. Dixon, “Determination of the fixed bed wall heat transfer coefficient using computational fluid dynamics”, *Numerical Heat Transfer, Part A Applications*, vol. 29, no. 8, pp. 777–794, 1996.
- [35] A. G. Dixon, M. Nijemeisland, and E. H. Stitt, “Systematic mesh development for 3d cfd simulation of fixed beds: Contact points study”, *Computers & Chemical Engineering*, vol. 48, pp. 135–153, 2013.
- [36] M. Kuroki, S. Ookawara, D. Street, and K. Ogawa, “High-fidelity cfd modeling of particle-to-fluid heat transfer in packed bed reactors”, *ECCE-6, Copenhagen*, pp. 16–21, 2007.
- [37] K. Murphy, *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning series, MIT Press, 2012.
- [38] D. A. Narciso and F. Martins, “Application of machine learning tools for energy efficiency in industry: A review”, *Energy Reports*, vol. 6, pp. 1181–1199, 2020.
- [39] K. Y. H. Lim, P. Zheng, and C.-H. Chen, “A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives”, *Journal of Intelligent Manufacturing*, pp. 1–25, 2019.
- [40] R. Cioffi, M. Travagliani, G. Piscitelli, A. Petrillo, and A. Parmentola, “Smart manufacturing systems and applied industrial technologies for a sustainable industry: A systematic literature review”, *Applied Sciences*, vol. 10, no. 8, p. 2897, 2020.
- [41] C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, 2018.
- [42] S. Sheehan and Y. Song, “Deep learning for population genetic inference”, *PLOS Computational Biology*, vol. 12, p. e1004845, 03 2016.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [44] S. Skansi, *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*. Undergraduate Topics in Computer Science, Springer International Publishing, 2018.
- [45] E. Coumans, “Bullet physics simulation”, in *ACM SIGGRAPH 2015 Courses*, p. 1, ACM SIGGRAPH, 2015.
- [46] D. Baraff, “An introduction to physically based modeling: rigid body simulation i—unconstrained rigid body dynamics”, *SIGGRAPH course notes*, vol. 82, 1997.
- [47] D. Baraff, “An introduction to physically based modeling: rigid body simulation ii—nonpenetration constraints”, *SIGGRAPH course notes*, pp. D31–D68, 1997.
- [48] F. Radjai and V. Richefeu, “Contact dynamics as a nonsmooth discrete element method”, *Mechanics of Materials*, vol. 41, no. 6, pp. 715–728, 2009.
- [49] “Bullet real-time physics simulation”. <https://pybullet.org/wordpress>, Sep 2020.
- [50] C. T. DeGroot and A. G. Straatman, “A finite-volume model for fluid flow and nonequilibrium heat transfer in conjugate fluid-porous domains using general unstructured grids”, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 60, no. 4, pp. 252–277, 2011.

- [51] K. Daizo and S. Motoyuki, “Particle-to-fluid heat and mass transfer in packed beds of fine particles”, *International Journal of Heat and Mass Transfer*, vol. 10, no. 7, pp. 845–852, 1967.
- [52] N. Wakao, S. Kaguei, and T. Funazkri, “Effect of fluid dispersion coefficients on particle-to-fluid heat transfer coefficients in packed beds: correlation of nusselt numbers”, *Chemical engineering science*, vol. 34, no. 3, pp. 325–336, 1979.
- [53] “Tensorflow.org”. <https://www.tensorflow.org>.
- [54] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.
- [55] M. Hardt, B. Recht, and Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent”, in *International Conference on Machine Learning*, pp. 1225–1234, PMLR, 2016.

## A. *Script* per la creazione della geometria in Blender

```
import bpy
import math
import random
import numpy as np
import scipy.stats as stats
#-----#
##### DATI IN INGRESSO: #####
Dp_medio =
dev_standard =
csv_file = r"C:\DATI\PROGRAMMI\BlenderData\PACKS\*.csv"
#-----#

rAve=Dp_medio*0.5
xSphereMax = 4           #numero di sfere sull'asse x
ySphereMax = 4           #numero di sfere nell'asse y
sphereFloors = 100      #numero di piani di sfere
sp = 3*rAve              #spazio tra una sfera e l'altra
offset = -4.2*rAve      #offset per la posizione x, y delle
    sfere
zoffset = 11            #offset in direzione z

Vave= 4.18879*rAve*rAve*rAve

N_sfere = xSphereMax*ySphereMax*sphereFloors+1
N_sfere = xSphereMax*ySphereMax*sphereFloors+1
lower = 0.12             #limite inferiore raggio [x0.1 mm]
upper = 0.825           #limite superiore raggio [x0.1 mm]
volume_list = np.random.normal(Vave, dev_standard, int(1.1*
    N_sfere)) #costruzione della distribuzione
radius_list = np.cbrt(volume_list/4.18879)
radius_list = radius_list[ (radius_list >= lower) & (radius_list
    <= upper)
random.shuffle(radius_list)
random.shuffle(radius_list)
random.shuffle(radius_list)

r=np.array(radius_list , dtype=float)
```

```
bpy.context.scene.rigidbody_world.steps_per_second = 60
bpy.context.scene.rigidbody_world.solver_iterations = 10
spheres_collection = D.collections.new('spheres')
C.scene.collection.children.link(spheres_collection)
C.view_layer.active_layer_collection = \
C.view_layer.layer_collection.children['spheres']

#-----#
##### POSIZIONAMENTO DELLE SFERE: #####
#-----#
a = 1
for z in range(0,sphereFloors):
    for x in range(0,xSphereMax):
        for y in range(0,ySphereMax):
            bpy.ops.mesh.primitive_uv_sphere_add(radius=r[a],
            location=(x*sp+offset, y*sp + offset, z*sp+
            zoffset))
            bpy.ops.object.shade_smooth()
            bpy.ops.object.randomize_transform(random_seed =
            random.uniform(0,1000), loc=(rAve, rAve, rAve),
            scale=(1, 1, 1))
            bpy.ops.rigidbody.object_add(type='ACTIVE')
            bpy.ops.rigidbody.mass_calculate(density=1.0)
            bpy.ops.rigidbody.shape_change(type='SPHERE')
            bpy.context.object.rigid_body.use_margin = False
            bpy.context.object.rigid_body.friction = 0.5
            bpy.context.object.rigid_body.linear_damping = 0.1
            bpy.context.object.rigid_body.restitution = 0.25
            a = a + 1

for item in bpy.data.collections['spheres'].all_objects:
    item.select_set(True)

bpy.ops.rigidbody.bake_to_keyframes(frame_start=0, frame_end
=250, step=5)
bpy.context.scene.frame_set(250)

#-----#
##### ESTRAZIONE DEI DATI: #####
#-----#
f=open(csv_file, 'w', newline='\n')
f.writelines(str('radius')+';'
+str('x')+';'
+str('y')+';')
```

```
+str('z')+ '\n')

for item in bpy.data.collections['spheres'].all_objects:
    f.writelines( str(round(0.5*item.dimensions[0], 3)) +' ;\n'
    ,
    +str(item.location[0])+' ;\n'
    +str(item.location[1])+' ;\n'
    +str(item.location[2])+' \n')

f.close()
```



## B. *Script* per la modifica dei punti di contatto in Salome

```
import csv
import salome
salome.salome_init()
theStudy = salome.myStudy
gg = salome.ImportComponentGUI("GEOM")
import GEOM
from salome.geom import geomBuilder
geompy = geomBuilder.New()
O = geompy.MakeVertex(0, 0, 0)
OX = geompy.MakeVectorDXDYDZ(1, 0, 0)
OY = geompy.MakeVectorDXDYDZ(0, 1, 0)
OZ = geompy.MakeVectorDXDYDZ(0, 0, 1)

#-----#
##### DEFINIZIONE DEI PARAMETRI: #####
#-----#
DpAve = #diametro medio particelle
inputFileName = "\*.csv" #percorso del file input
contactDistance = 0.001 #distanza massima per la definizione
    dei punti di contatto [x0.1 mm]
rb_dp = 0.05 #rapporto raggio bridges su diametro
    medio particelle
NDpx=4 #numero di diametri in direzione x
NDpy=4 #numero di diametri in direzione y
NDpz=8 #numero di diametri in direzione z

#-----#
##### GENERAZIONE DELLE SFERE: #####
#-----#
radiusAve = DpAve*0.5
xmax=NDpx*radiusAve
ymax=NDpy*radiusAve
tzmax=NDpz*(2*radiusAve)

print("Importing\spheres\dots")
rem=0
vertexes = []
spheres = []
```

```

centres = []
radii = []
with open(inputFileName, 'rt') as csvFile:

    csvReader = csv.reader(csvFile, delimiter=';')

    thisIsHeader = True
    for row in csvReader:

        if thisIsHeader == True:

            if "'radius'" in str(row) and "'x', 'y', 'z'" in str
                (row):
                    radiusColumnIndex = row.index('radius')
                    xColumnIndex = row.index('x')
                    yColumnIndex = row.index('y')
                    zColumnIndex = row.index('z')
                    thisIsHeader = False
            else:
                x = float(row[xColumnIndex])
                y = float(row[yColumnIndex])
                z = float(row[zColumnIndex])

                if -(xmax+2*radiusAve) < x < (xmax+2*radiusAve) and
                    -(ymax+2*radiusAve) < y < (ymax+2*radiusAve) and
                    -(2*radiusAve) < z < (tzmax+2*radiusAve):
                    radius = float(row[radiusColumnIndex])
                    radii.append(radius)
                    vertex = geompy.MakeVertex(x, y, z)
                    vertexes.append(vertex)
                    sphere = geompy.MakeSpherePntR(vertex, radius)
                    spheres.append(sphere)
                    centres.append(vertex)
                else:
                    rem=rem+1

nbSpheres = len(spheres)
print(nbSpheres, "Spheres created")

##### GENERAZIONE DEI BRIDGES: #####
import math
rb = rb_dp * DpAve

```

```
arcoseno = math.asin(rb/radiusAve)
coseno = math.cos(arcoseno)
cap = radiusAve * (1 - coseno )

bridges = []
extrudedplanes = []

print("Generating sphere-sphere bridges ... ")
for i in range(nbSpheres):

    sphere1 = spheres[i]
    radius1 = radii[i]
    if radius1>rb:
        cap1 = radius1*(1-math.cos(math.asin(rb/radius1)))
    else:
        cap1=cap
    for j in range(nbSpheres):

        if i > j:
            sphere2 = spheres[j]
            radius2 = radii[j]
            distance = ( geompy.MinDistance(vertexes[j],
                vertexes[i]) - (radius1+radius2))

            if distance <= contactDistance:

                if radius2>rb:
                    cap2 = radius2*(1-math.cos(math.asin(rb/
                        radius2)))
                else:
                    cap2=cap
                normalVector = geompy.MakeVector(vertexes[i],
                    vertexes[j])
                p1=vertexes[i]
                p2=vertexes[j]
                contactPoint = geompy.
                    MakeTranslationVectorDistance(p1,
                    normalVector, radii[i]+0.5*distance)
                cil1 = geompy.MakeCylinder(p1, normalVector,
                    radiusAve, 0.1)
                cilt1 = geompy.MakeTranslationVectorDistance(
                    cil1, normalVector, (0.995*radius1-cap1))
                cil2 = geompy.MakeCylinder(p2, normalVector,
                    radiusAve, 0.1)
```

```
cilt2 = geompy.MakeTranslationVectorDistance(
    cil2 , normalVector , -(0.1+0.995*radius2-cap2)
)
sphere1 = geompy.MakeCut(sphere1 , cilt1)
sphere2 = geompy.MakeCut(sphere2 , cilt2)
h = 0.5*radiusAve
bridge = geompy.MakeCylinder(contactPoint ,
    normalVector , rb , h)
bridge1 = geompy.MakeTranslationVectorDistance(
    bridge , normalVector , -h/2)
bridges.append(bridge1)
spheres[j] = sphere2
spheres[i] = sphere1
print ("\ncontact_points_sphere/sphere_=", len(bridges))

totspheres = geompy.MakeCompound(spheres)
geompy.addToStudy(totspheres , "totspheres")
bridges_SS = geompy.MakeCompound(bridges)
geompy.addToStudy(bridges_SS , "bridges_sphere/sphere")
cut_BS = geompy.MakeCut(bridges_SS , totspheres)
geompy.addToStudy(cut_BS , "cut_BS")
print ("bridges_generated")

#-----#
##### GENERAZIONE DELLA GEOMETRIA FINALE: #####
#-----#
totGeom = geompy.MakeFuse(totspheres , cut_BS)
geompy.addToStudy(totGeom , "totGeom")
print ("final_geometry_generated")
```

## Lista dei simboli

Di seguito viene riportata la lista dei simboli utilizzati nel presente documento, insieme alla relativa unità di misura. I pedici  $f, s$  indicano rispettivamente il dominio fluido e il dominio resina.

Simbolo	Definizione	Unità di misura
$\alpha_{f,s}$	Diffusività termica	$m^2/s$
$\Gamma$	Coefficiente di trasporto generico	$m^2/s$
$\frac{\Delta P}{L}$	Caduta di pressione lineare	$Pa/m$
$\Delta T_{ref}$	Differenza di temperatura di riferimento per il calcolo del coefficiente di scambio termico	$K$
$\Delta t$	Intervallo di tempo	$s$
$\delta$	Coefficiente di condizionamento dell'algoritmo "adam"	-
$\delta_n$	Distanza tra due corpi rigidi	$m$
$\epsilon$	Porosità esterna del letto	-
$\lambda$	<i>Learning rate</i>	-
$\lambda_{adj}$	<i>Learning rate</i> corretto (algoritmo "adam")	-
$\mu$	Viscosità dinamica	$Pa\ s$
$\mu_a$	Coefficiente di attrito	-
$\rho_{1,2}$	Coefficienti di decadimento dell'algoritmo "adam"	-
$\rho_{f,s}$	Densità	$kg/m^3$
$\tau$	Momento torcente	$N\ m$
$\phi$	Proprietà generica trasportata	Variabile
$\omega_i$	Velocità angolare della sfera i-esima	$rad/s$
$A$	Area interfacciale	$m^2$
$A_i$	Valore mediato esponenzialmente del peso i-esimo	-
$a_{f,s}$	Area interfacciale specifica	$m^{-1}$
$b$	<i>bias</i> del neurone/ <i>layer</i>	-
$C_p$	Calore specifico a Pressione costante	$J/(kg\ K)$
$c_i$	Posizione del centro di massa della sfera i-esima	$m$
Co	Numero di Courant	-
$\bar{D}_p$	Diametro medio delle particelle	$\mu m$
$d$	Coefficiente di Darcy	$m^2$

$d_c$	Tolleranza sulla distanza di contatto	$\mu m$
$d_{i,j}$	Distanza tra i centri delle sfere $i$ e $j$	$\mu m$
$dv_{50}$	Momento 4,3 della distribuzione dei diametri delle particelle	$\mu m$
$e$	Numero di <i>epochs</i>	-
$e_n$	Coefficiente di restituzione normale	-
$e_t$	Coefficiente di restituzione tangenziale	-
$\mathbf{F}$	Forza	$N$
$F_i$	Valore mediato esponenzialmente della derivata parziale della funzione di perdita lungo la direzione $i$	Variabile
$f_{n,t}$	Componenti normale e tangenziale della forza di contatto	$N$
$f_{loss}$	Funzione di perdita della rete neurale	Variabile
$\hat{H}_{f,s}$	Entalpia specifica	$J/kg$
$h_{fs}$	Coefficiente di scambio termico	$W/(m^2 K)$
$\mathbf{I}$	Tensore del momento di inerzia	$kg m^2$
$\mathbf{J}$	Impulso	$N s$
$k_{bed}$	Permeabilità del letto	$m^{-2}$
$k_{eff}$	Conducibilità termica efficace dei <i>bridges</i>	$W/(m K)$
$k_{f,s}$	Conducibilità termica	$W/(m K)$
$L_i$	Momento angolare della sfera $i$ -esima	$(kg m^2)/s$
$M$	Massa	$kg$
$N_{diff}$	Numero di diffusione	-
$\mathbf{n}$	Versore normale	$m$
$P_i$	Quantità di moto della sfera $i$ -esima	$(kg m)/s$
$p$	Pressione	$Pa$
$\bar{Q}_{vol}$	Densità di potenza termica scambiata	$W/m^3$
$q$	Flusso di calore	$W/m^2$
$R$	Raggio della sfera	$m$
$r_b$	Raggio dei <i>bridges</i>	$\mu m$
$Re_p$	Numero di Reynolds riferito alla particella	-
$S_\phi$	Termine sorgente della proprietà $\phi$	
$T$	Temperatura	$K$
$\bar{T}_{f,s}$	Temperatura mediata sul volume	$K$
$T_{min}$	Temperatura minima incontrata nel sistema	$K$
$\mathbf{u}$	Vettore velocità	$m/s$
$\mathbf{u}^-, \mathbf{u}^+$	Velocità della sfera prima e dopo l'urto	$m/s$
$u_{n,t}$	Componenti normale e tangenziale della velocità relativa tra due corpi rigidi	

$u_{sup}$	Velocità superficiale lungo la direzione principale del flusso	$m/s$
$V$	Volume	$m^3$
$V_{f,s}$	Volume del dominio	$m^3$
$v_i$	Velocità del centro di massa della sfera $i$ -esima	$m/s$
$w$	Vettore dei pesi della rete neurale	-
$x$	Vettore delle <i>features</i>	Variabile
$Y$	Vettore di stato	Variabile
$y$	Vettore dei <i>labels</i>	Variabile
$\hat{y}$	Vettore delle previsioni della rete	Variabile



## Acronimi

<b>AI</b>	<i>Artificial Intelligence</i>
<b>ANN</b>	<i>Artificial Neural Network</i>
<b>CD</b>	<i>Contact Dynamics</i>
<b>CDS</b>	<i>Central Differencing Scheme</i>
<b>CFD</b>	<i>Computational Fluid Dynamics</i>
<b>CFL</b>	Courant-Friedrichs-lewy (Condizione di)
<b>CPU</b>	<i>Central Processing Unit</i>
<b>CSV</b>	<i>Comma Separated Values</i>
<b>DBC</b>	<i>Dynamic Bounding Capacity</i>
<b>DEM</b>	<i>Discrete Elements Method</i>
<b>DNS</b>	<i>Direct Numerical Simulation</i>
<b>DT</b>	<i>Digital Twin</i>
<b>FVM</b>	<i>Finite Volumes Method</i>
<b>HIC</b>	<i>Hydrophobic Interaction Chromatography</i>
<b>IEC</b>	<i>Ion Exchange Chromatography</i>
<b>LUDS</b>	<i>Linear Upwind Differencing Scheme</i>
<b>MAE</b>	<i>Mean Absolute Error</i>
<b>ML</b>	<i>Machine Learning</i>
<b>MSE</b>	<i>Mean Squared Error</i>
<b>PDE</b>	<i>Partial Differential Equation</i>
<b>QbD</b>	<i>Quality by Design</i>
<b>RAM</b>	<i>Random Access Memory</i>
<b>ReLU</b>	<i>Rectified Linear Unit</i>
<b>REV</b>	<i>Representative Elementary Volume</i>
<b>SEC</b>	<i>Size Exclusion Chromatography</i>
<b>SGD</b>	<i>Stochastic Gradient Descent</i>
<b>SIMPLE</b>	<i>Semi-Implicit Method for Pressure-Linked Equations</i>
<b>STL</b>	<i>Standard Triangle Language</i>



## Ringraziamenti

Arrivato al termine di questo percorso, colgo l'occasione per guardarmi un po' indietro e, per quel che è possibile, rendere conto dei loro meriti alle persone che mi hanno affiancato e accompagnato.

Ringrazio tutti coloro che mi hanno permesso di far parte di GSK Vaccines, e in particolare Antonio, che mi ha ascoltato, supportato e consigliato lungo tutta la durata di un'esperienza che per me ha avuto grande valore. Ringrazio i prof. Daniele Marchisio e Gianluca Boccardo, insieme alla dottoressa Agnese Marcato, per i loro preziosi consigli, gli stimoli e l'incoraggiamento a lavorare in autonomia che mi hanno dato, mettendomi alla prova e rendendo appassionante il mio percorso.

Ringrazio tutti i fratelli e sorelle adottivi e gli amici che ho seminato in giro per l'Italia e oltre nel corso della mia carriera universitaria, il bene che ho ricevuto e quello che ho regalato è stato ed è il sale dei miei anni, ciò che ha reso gustosa e magnifica questa esperienza. Con l'augurio che sia così per sempre, confesso che coltivo il sogno di rincontrarvi tutti insieme, come la grande famiglia adottiva che siete stati per me. Nell'attesa, buona vita a tutti!

Ringrazio i miei genitori, Sandra e Paolo, perché hanno sempre creduto in me e nelle mie scelte, è grazie alla vostra fiducia che ho trovato la mia. Ringrazio mia sorella Elena, perché la sua determinazione e la sua sensibilità sono state una guida durante tutta la mia vita; mi hai fatto crescere, tanto.

Ringrazio Giulia, che per me è più di una compagna di vita perché con lei ho condiviso la nostra crescita personale, a partire dalle piccole cose che, coltivate, sono diventate grandi e ora definiscono chi siamo, e i nostri sogni e aspirazioni. È una fortuna compiere questo viaggio insieme a te, ti amo tanto!