UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

DOCTORATE SCHOOL
INDUSTRIAL INNOVATION ENGINEERING
XXXVIII CYCLE

MULTISCALE DESIGN AND OPTIMISATION APPROACHES FOR THERMAL MANAGEMENT SYSTEMS OF BATTERY ELECTRIC VEHICLES

Scientific Tutor:
Prof. Ing. Diego Angeli

Scientific Co-Tutor:
Prof. Ing. Marco Cavazzuti

Candidate:
Ludovico Campanelli

School Dean:
Prof. Ing. Franco Zambonelli

UNIVERSITY OF MODENA AND REGGIO EMILIA
ACADEMIC YEAR 2024/2025

Abstract

In battery electric vehicles (BEVs), various physical processes occur at different scales. For instance, in a battery system, heat transfer at full pack level depends on chemical reactions occurring within individual battery cells. In this work, multiscale models are presented to address the design and optimisation of thermal management systems in BEVs, leveraging their suitability for modelling the interactions between phenomena on different scales. The project was funded by the Sustainable Mobility Centre (MOST) as part of the National Recovery and Resilience Plan (PNRR), and part of the activities were also carried out in the framework of the European project SCAPE, which aims at standardising high efficiency power electronic units for sustainable vehicles.

In the first part of the work, a possible design for an integrated battery system coupled with a power converter unit through a liquid cooling plate is proposed. The study is conducted using a self-developed generalised toolbox for lumped parameter analysis of thermofluid systems. This toolbox exploits graph theory to represent the physical domain, where the balance equations are solved while ensuring consistency of measurement units. Source terms are expressed in linearised form, with possible non linearities managed by means of iterative procedures. Based on this framework, two solvers are implemented: a one dimensional fluid flow solver for pipe networks and a thermal network solver, which offers greater flexibility when the temperature is the only unknown to predict. The latter is employed to model the full assembly and is coupled with an electrochemical battery model, operating at the chemical reaction scale, and with a Computational Fluid Dynamics (CFD) model to estimate the heat transfer coefficient of the plate. This multiscale framework provides a good trade-off between accuracy and computational efficiency, reducing the computational cost with respect to full system simulations with higher-order methods.

In the second part of the work, the problem of the optimal design of a battery cooling plate

is addressed, building upon the case study presented in the first part. To minimise the mean temperature and improve temperature uniformity across the cooling plate, a metaheuristic algorithm is presented to optimise the channel layout. These optimised layouts are generated using a reduced-order optimisation approach that combines the thermal network solver with Ant Colony Optimisation (ACO), a method particularly suited for identifying optimal paths (*i.e.* channel layouts) within a graph. Although this procedure possesses limitations compared to high-order topology optimisation, it represents a computationally efficient and robust alternative for obtaining optimal solutions that are easily manufacturable or can be utilised as initial configurations for high-order optimisations.

Beyond battery cooling, the thermal management system of BEVs also includes the HVAC system to ensure good air quality and comfortable thermohygrometric conditions for passengers. In the last part of the thesis, a multiscale methodology for HVAC design is proposed, focusing on infection risk due to airborne exhaled droplets, which depends on local thermohygrometric conditions of the environment. Droplet dispersion is analysed through a multiscale model combining CFD simulations with an analytical model: the CFD model captures the macroscopic flow features of the exhalation, while the analytical model uses the CFD results to evaluate microscopic droplet dynamics. Therefore, this approach exploits the capability of CFD to predict flow structures, as well as the flexibility and accuracy of the analytical model used to evaluate droplet trajectories. The method is applied to a simple base case to demonstrate its potential and provide insight into transmission risk. This approach can be easily extended to vehicle ventilation studies, highlighting its relevance for HVAC design in BEVs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increasing concentration of Greenhouse Gases (GHG) in the atmosphere is placing pressure on the global community to adopt effective mitigation actions. As synthesized in the Sixth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC) [1], human influence has unequivocally warmed the atmosphere, increasing the global surface temperature by 1.1 °C with respect to the level during 1850–1900. To address this trajectory, the international community adopted the Paris Agreement [2], establishing a binding obligation to hold the increase in the global average temperature to well below 2 °C (preferably 1.5 °C) above pre-industrial levels. In the European context, these commitments are legally codified in the European Climate Law (Regulation (EU) 2021/1119) [3], which mandates climate neutrality by 2050 and a net emission reduction of at least 55% by 2030 with respect to 1990 levels. These legislative constraints dictate the technological roadmap for the transport sector, necessitating a structural departure from fossil-fuel-based propulsion.

The transport sector remains a critical focal point for these decarbonisation efforts, accounting for approximately one-quarter of the total greenhouse gas emissions within the European Union [4]. Unlike other industrial sectors which have achieved substantial emission reductions, transport has historically exhibited increasing emission trends. This trend is depicted in Figure 1.1, where the relative variation of GHG emission in EU compared from 1990 levels is plotted for different sectors. The plot highlights how during the years the transport sector (red curve) is the only sector with higher emissions with respect to 1990; the years affected by the SARS-CoV-2 pandemic were an exception to that trend. The aviation sector was particularly affected by the pandemic, with international aviation emissions falling by 58% between 2019 and 2020. However, this reduction was only temporary. Air traffic emissions increased by 25% in 2021 and by a further 57% in 2022. It is important to point that the data before the emissions during 2020, refers to 2015. For this reason from the plot

it seems that the negative trend due to COVID-19 pandemic started in 2015, rather than 2020. However this is only an artifact caused by the linear interpolation between the 2015 and 2020 data points.

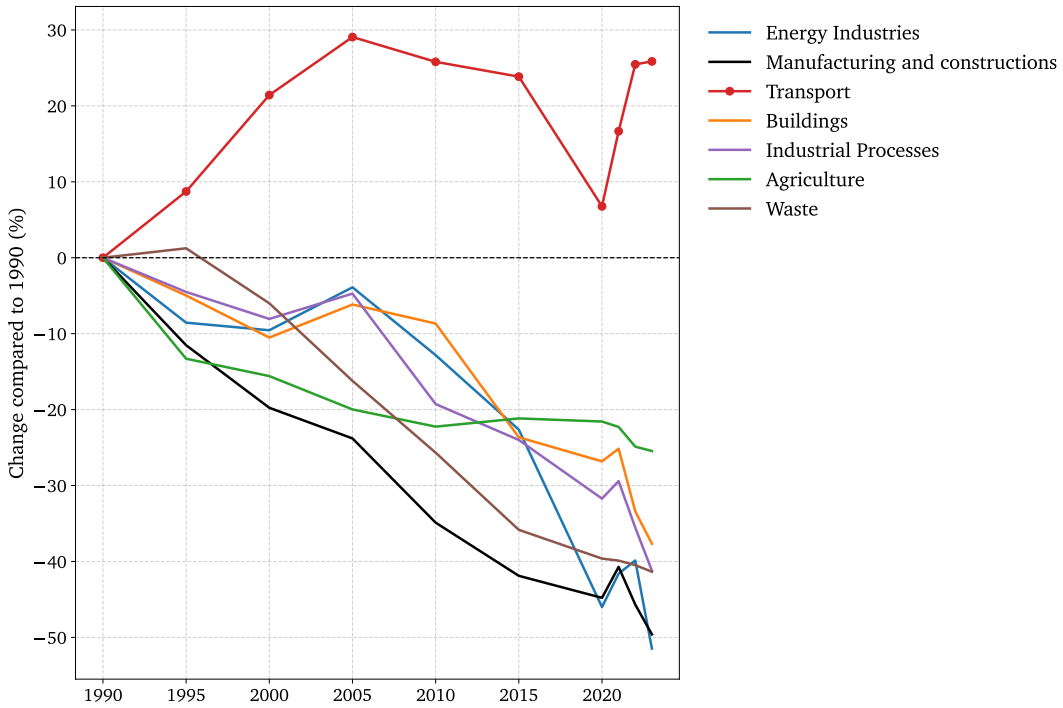


Figure 1.1: Relative variation from 1990 of GHG emissions in EU by sector [5].

A substantial contribution of transport emissions originates from road transport. As depicted in the top chart in Figure 1.2, in 2023 the road transport is responsible for 73% EU's transport GHG emissions [6]. The greatest portion of road transport emissions is due to cars, which is responsible 60% of emissions (bottom chart in Figure 1.2).

The "Fit for 55" legislative package has established stringent regulatory frameworks to mitigate these impacts [7]. Central to this overhaul is Regulation (EU) 2023/851 [8], which amends the previous performance standards for CO_2 emissions. This regulation enforces a trajectory towards zero-emission mobility, explicitly requiring a 100% reduction in CO_2 emissions for new cars registered from 2035 onwards. This requirement has accelerated the transition to electrified powertrains, necessitating rapid technological advancements in energy storage and propulsion systems.

Vehicle electrification strategies are categorised based on the energy source and the electromechanical interaction within the powertrain. Hybrid Electric Vehicles (HEVs) represent the initial step in this spectrum, utilising an internal combustion engine as the primary power source. The engine is supported by an electric motor and a small battery pack. In these architectures, electrical energy is generated exclusively through regenerative braking [9] or by the engine acting as a generator, without the capability to recharge from the electrical

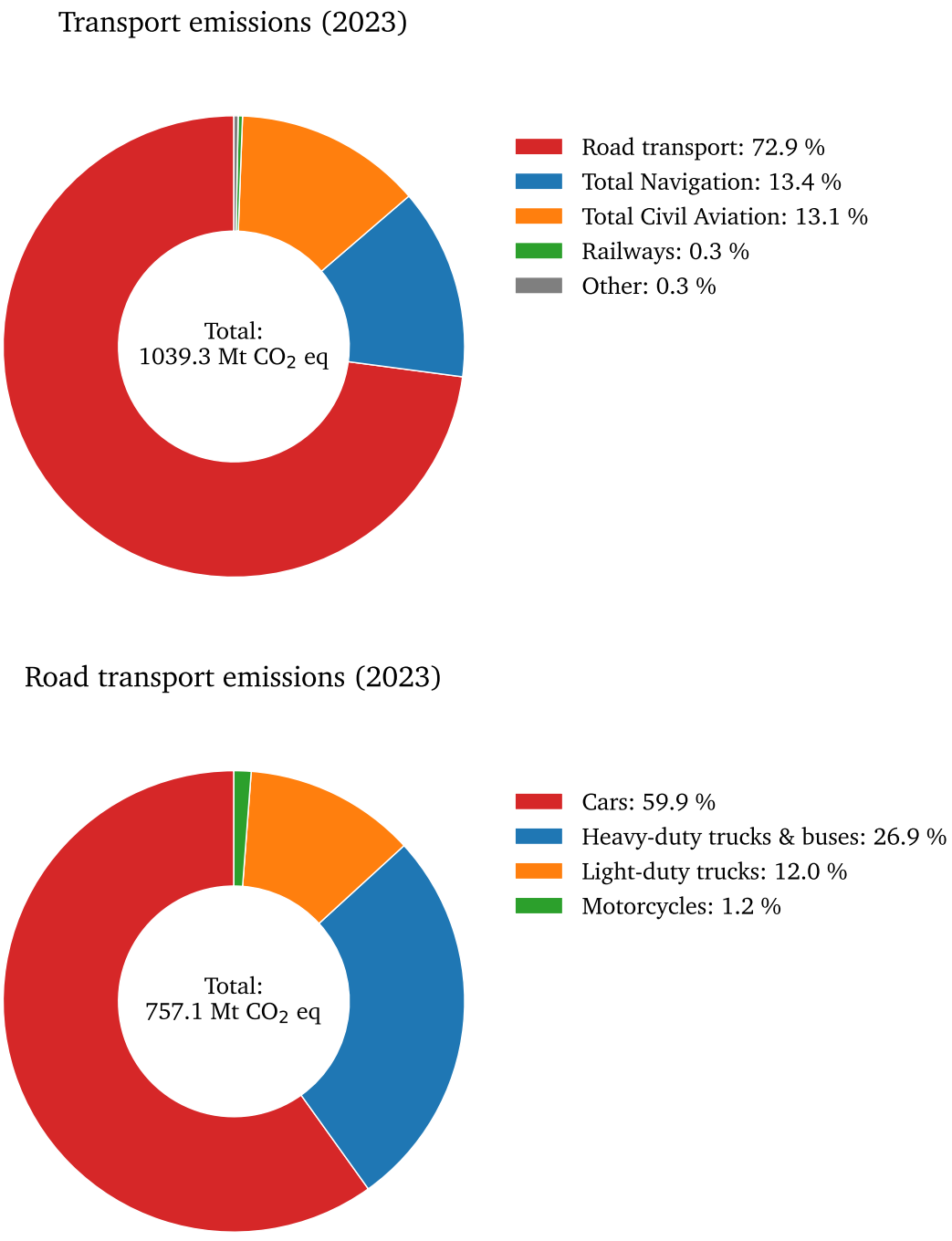


Figure 1.2: Emissions in EU during 2023 by transport sector [6].

grid. The battery capacity in HEVs is typically limited, generally remaining below 2 kWh, acting primarily as a power buffer for load levelling to maximize system efficiency [10]. The integration of the electric motor provides greater flexibility to the powertrain, as its characteristics differ significantly from those of internal combustion engines (ICEs). Unlike an ICE, an electric motor can instantly deliver maximum torque from 0 rpm and operates with substantially higher efficiency ($\sim 90\%$). Furthermore, because they contain significantly fewer moving components, electric motors require less maintenance, are inherently lighter, and ensure smoother operation with low levels of noise and vibration. Consequently, their compact size provides greater packaging flexibility within the vehicle. Given the small size of the Energy Storage System (ESS), in HEV the electric mode is activated only during low speed driving and transient phases such as stop-and-run driving pattern, which are frequent inside cities. This not only increases the overall efficiency of the power train reducing emissions, but ensure also a better driving comfort [11].

A higher degree of electrification is observed in Plug-in Hybrid Electric Vehicles (PHEVs), which integrate a larger battery pack and an onboard charger, allowing connection to the external electrical grid to charge to ESS. PHEVs serve as a bridge technology, capable of operating in charge-depleting mode (pure electric) for moderate distances and charge-sustaining mode (hybrid) for longer journeys [12]. Typical battery capacities for PHEVs range from 10 kWh to 25 kWh, depending on the vehicle segment and the targeted all-electric range [13].

Alternative zero-emission solutions include Fuel Cell Electric Vehicles (FCEVs), which utilise a Proton Exchange Membrane Fuel Cell (PEMFC) to convert the chemical energy of stored hydrogen into electricity. Fuel cells generate electricity from chemical reactions, thus do not burn fuel. This means that they do not create pollutants and produce much less heat than ICEs [14]. FCEVs offer the advantage of long driving ranges and short refuelling times comparable to other electric vehicles. In these systems, the fuel cell stack is the primary energy source, but it is almost invariably hybridised with a secondary energy storage system, typically a small battery pack (1–5 kWh) [15] or supercapacitors. This buffer storage is essential to handle high-power transients during acceleration and to recover kinetic energy during braking, which the fuel cell cannot absorb. While FCEVs look promising for heavy-duty applications, the passenger car market is currently trending towards battery-dominant architectures due to their higher infrastructure maturity and well-to-wheel efficiency [16].

Consequently, the most significant paradigm shift is embodied by Battery Electric Vehicles (BEVs), which rely exclusively on chemical energy stored in a high-voltage battery pack to power the traction motor, therefore without the presence of the ICE, thus avoiding the emissions of pollutant. According to the International Energy Agency, in 2022 about 70% of the global stock of electric cars were BEVs [17]. As the sole energy source for propulsion and auxiliary systems, the battery pack in BEVs requires a substantially higher capacity

compared to hybrid or fuel cell architectures. Contemporary passenger BEVs are equipped with high-energy-density lithium-ion storage systems typically ranging from 30 kWh for compact city cars up to 100 kWh or more for long-range luxury vehicles.

The shift to such high-energy systems introduces complex engineering challenges, primarily regarding the thermal sensitivity of lithium-ion electrochemistry. The performance, safety, and longevity of these systems are intrinsically linked to their thermal behaviour [18]. Lithium-ion cells can generally operate between −20 and 60 °C, but their optimal temperature window is limited in the range 15–35 °C. Operating conditions outside optimal temperature window lead to rapid degradation [19]. Low temperatures result in increased internal impedance and the risk of lithium plating, which irreversibly reduces capacity [20]. Conversely, high temperatures accelerate the growth of the Solid Electrolyte Interphase (SEI) layer and can trigger thermal runaway, a catastrophic failure mode characterised by a self-sustaining exothermic reaction [21]. Therefore, the design and optimisation of Battery Thermal Management Systems (BTMSs) has emerged as critical research area to ensure the reliability of battery electric vehicles under varying operating conditions.

1.1. Li-ion batteries

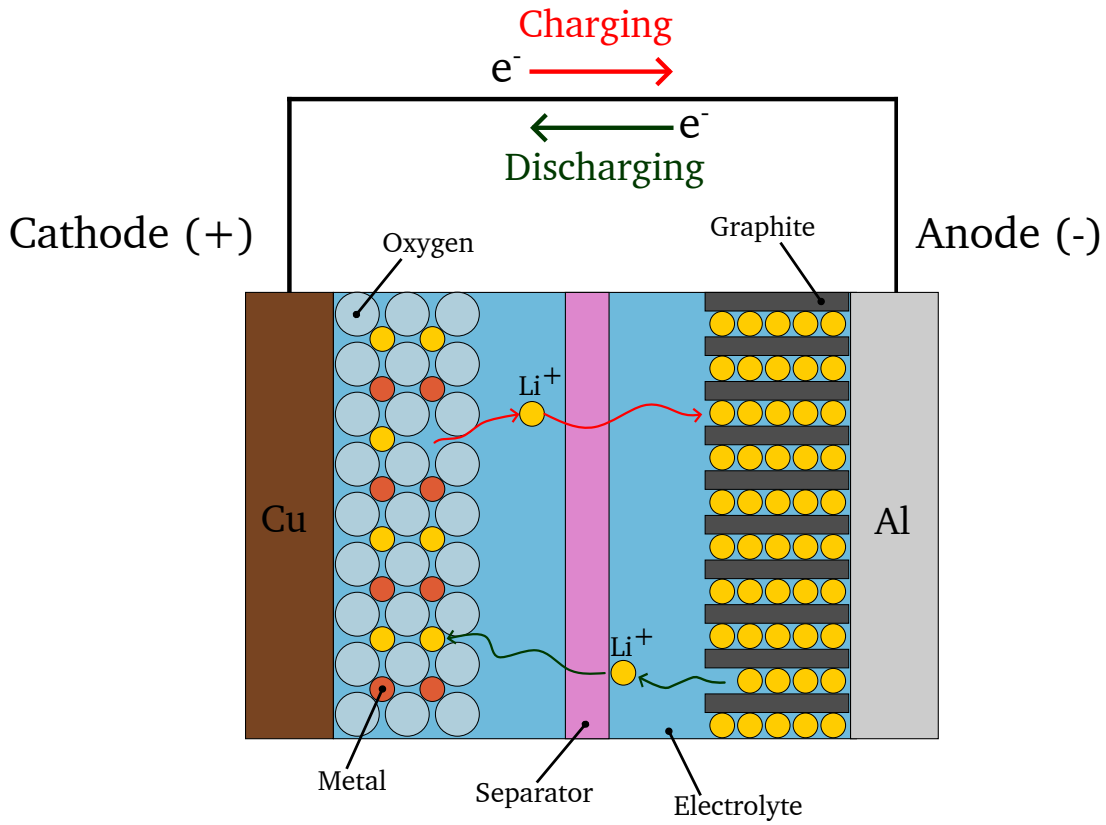


Figure 1.3: Charging and discharging process of a Li-ion battery.

The battery constitutes the primary energy storage system in a BEV, serving as the determining factor for the vehicle's range, cost, and overall weight [22]. While the automotive

industry has explored various chemistries, ranging from the heavy and low-energy Lead-Acid units ($30-50\,Wh/kg$) to the Nickel-Metal Hydride (Ni-MH) batteries used in early hybrids, the Lithium-Ion Battery (LIB) has established itself as the standard for modern propulsion.

Lithium-ion batteries are favoured for their compact size and low weight, offering a specific energy in the range of 100 to $180\,Wh/kg$ and operating at a typical nominal cell voltage of approximately $4\,V$ [14] (compared to $1.2\,V$ for standard Ni-MH cells). Despite these advantages, Li-ion batteries present several challenges: high manufacturing costs, strict operating temperature limits, and a dependence on the availability of critical raw materials.

The fundamental mechanism of a lithium-ion battery involves the movement of lithium ions (Li^+) between the electrodes through a separator. During the discharge cycle, lithium ions migrate from the negative electrode to the positive electrode, while the reverse process occurs during charging.

Figure 1.3 illustrates the charging and discharging process of a Li-ion battery. In a typical configuration, the negative electrode is composed of carbon, with graphite being the most commercially common material. The positive electrode consists of a lithium-containing compound, which is generally selected from three material classes: a layered oxide (e.g., Lithium Cobalt Oxide, $LiCoO_2$), a polyanion (e.g., Lithium Iron Phosphate, $LiFePO_4$), or a spinel (e.g., Lithium Manganese Oxide, $LiMn_2O_4$) [21]. The ion transport is facilitated by the electrolyte, which consists of a solution of lithium salt in a non-aqueous solvent such as ethylene carbonate or diethyl carbonate. The current collectors serve to conduct electrons from the active materials to the external electrical circuit and are distinct for each electrode: copper (Cu) is used for the negative electrode, while aluminium (Al) is used for the positive electrode [21]. During discharging, the lithium ions move from the anode to the cathode, passing through the electrolyte and crossing the separator. This ion movement is driven by an external potential difference across the battery terminals. When the ions arrive at the cathode, they intercalate between the cathode material layers and remain there since they reach a thermodynamically stable state. At this point, a non-equilibrium state between charges arises because the electrons cannot follow the positive ions, as the electrolyte and separator act as electrical insulators. Consequently, to restore equilibrium, the electrons must reach the cathode by flowing through the external circuit. Heat is generated during both ion diffusion in the electrolyte and electron movement. Taking the specific example of a $LiMn_2O_4$/graphite configuration, the electrochemical reactions occurring at the electrode/electrolyte interfaces are described as follows [21]:

$$\text{Positive Electrode:} \quad Li_{y-x}Mn_2O_4 + xLi^+ + xe^- \underset{\text{charge}}{\overset{\text{discharge}}{\rightleftarrows}} Li_yMn_2O_4 \tag{1.1}$$

$$\text{Negative Electrode:} \quad Li_xC_6 \underset{\text{charge}}{\overset{\text{discharge}}{\rightleftarrows}} C_6 + xLi^+ + xe^- \tag{1.2}$$

Figure 1.4 shows the three different configurations usually employed for battery cells. The difference between cylindrical and prismatic cells lies primarily in their geometry. Cylindrical cells are typically smaller; therefore, for a given volume, a pack containing cylindrical batteries can hold a greater number of cells compared to a pack of prismatic cells. However, cylindrical cells do not tessellate perfectly, resulting in interstitial air gaps. While these gaps can aid thermal management, prismatic cells allow for a higher volumetric efficiency (calculated as the total volume of cells divided by the volume of the pack) and, consequently, higher energy density. Furthermore, the selection is influenced by the BTMS; for instance, prismatic cells generally offer a larger surface area for heat transfer, although this is not a universal rule. In general, cylindrical cells offer a longer cycle life and a lower risk of leakage due to their round shape, which results in better pressure distribution within the cell. Pouch cells possess the highest energy density as they are enclosed in a flexible foil rather than a rigid case, resulting in a non-rigid structure. Consequently, they require specialised enclosures when employed in EVs, as they are susceptible to mechanical damage. Pouch cells are not as widely adopted as cylindrical and prismatic cells; this is partly attributed to the manufacturing process, which presents inefficiencies that increase the manufacturing cost [23].

Figure 1.4: Li-ion battery common geometries; image readapted from [24].

1.1.1. Degradation Mechanisms in Li-ion Batteries

The operational performance of commercial lithium-ion batteries inevitably deteriorates over time as a result of irreversible physical and chemical changes. These alterations primarily manifest as capacity fade, defined as a decline in energy storage capability, and power fade, characterised by an increase in internal resistance. These observable symptoms stem from two principal degradation modes: Loss of Lithium Inventory (LLI), which arises from the consumption of Li-ions during side reactions, and Loss of Active Material (LAM), which results from structural damage or particle isolation. Although these mechanisms often occur concurrently, they can be classified into electrochemical, mechanical, and thermal processes.

Electrochemical Degradation Electrochemical degradation encompasses the chemical side reactions occurring at the electrode-electrolyte interfaces and is a primary driver of LLI and impedance growth. A critical factor in the electrochemical degradation of lithium-ion cells is the Solid Electrolyte Interphase (SEI), a passivation protective layer that forms spontaneously on the anode surface due to the thermodynamic instability of the graphite anode in organic electrolytes [25]. The SEI serves the critical function of preventing electron transfer to the electrolyte and protecting the anode from further reaction. Despite this protective role, the SEI inherently introduces several detrimental effects that compromise the cell's lifespan.

Specifically, the initial SEI forms during the first charge cycle, as the liquid electrolyte is chemically stable only within a voltage window of 0.8 to 4.5 V relative to lithium metal [26, 27], whereas the graphite anode operates at a significantly lower potential of approximately 0.05 V [28]. Consequently, during this initial charge (the formation period), the electrolyte is subjected to a potential below its stability threshold, triggering a cathodic reduction reaction. This decomposition mechanism irreversibly consumes active lithium ions and liquid solvent, degrading them into solid precipitates and gaseous byproducts. The resulting solid compounds deposit onto the anode surface to form a film that facilitates ionic transport while electrically insulating the electrolyte, thereby blocking further electron transfer and preventing continuous gas evolution.

However, the SEI is not a static structure; it continuously thickens during operation, a degradation process typically correlated with the square root of time [29]. Experimental investigations indicate that the thickness of the SEI layer ranges from 3 to 100 nm [30, 31], depending on the specific chemistry of the considered battery. The continuous growth of the SEI layer increases the internal impedance of the cell. As the internal resistance rises, the heat generated during battery operation increases due to the Joule effect, potentially accelerating thermally-driven degradation mechanisms.

While the SEI is considered the primary electrochemical degradation mechanism [32], several other electrochemical phenomena contribute to battery ageing:

i. *CEI*: similar to the SEI on the anode, a passivation layer known as the Cathode Electrolyte Interphase (CEI) forms on the positive electrode from redox reactions between the cathode and the electrolyte [33]. The CEI is typically much thinner than the SEI, ranging from 1 to 5 nm [34]. Although the CEI is generally considered more stable than the SEI, it acts as an electrically insulating layer that impedes Li^+ diffusion. This blockage leads to a rise in internal impedance, which directly contributes to increased heat generation during high-current operations and also leads to capacity fade.

ii. *Lithium plating*: it is a severe degradation mechanism where metallic lithium deposits on the anode surface instead of intercalating into the graphite structure. This occurs under conditions where the intercalation kinetics are sluggish, such as low temperatures,

high State of Charge (SOC), or high charging rates. Plating is thermodynamically possible when the anode potential drops below 0 V relative to lithium metal [35]. This reaction consumes active Li-ions and electrolyte, reducing the inventory available for charge and discharge cycles, which leads to additional capacity loss. The deposited lithium can react with the electrolyte to form a secondary SEI, consuming further lithium inventory. Furthermore, plating often occurs unevenly, potentially leading to dendritic growth that poses significant safety risks, including internal short circuits.

iii. *Li-stripping*: it refers to the re-oxidation of plated lithium during discharge, where the metallic lithium dissolves back into Li-ions. While theoretically reversible, this process is often incomplete. If the plated lithium loses electrical contact with the graphite electrode, often due to incomplete dissolution or mechanical fracture, it becomes electrochemically isolated. This isolated lithium (commonly referred to as 'dead lithium') can no longer participate in the electrochemical process, resulting in permanent LLI.

iv. *Corrosion*: corrosion of the current collectors leads to a loss of electrical contact and increased impedance. The aluminium collector on the cathode is protected by a native passivating layer, but it is susceptible to pitting corrosion at high potentials [36]. Conversely, the copper collector on the anode is vulnerable to dissolution during over-discharge events when the anode potential rises. The dissolved copper ions can migrate and re-deposit on the anode during subsequent charging, potentially forming dendrites. The corrosion products and the loss of contact area significantly increase the contact resistance within the cell, which becomes a localised source of heat generation during operation.

Mechanical and Thermal Degradation Mechanical degradation is primarily driven by the volume expansion and contraction of active materials during cycling. Graphite expands by roughly 10% [37], while silicon can expand up to 300% [38]. These repetitive volume changes induce stress that leads to particle fracture, including both intergranular and intragranular cracking. Cracking exposes fresh active material surfaces to the electrolyte, triggering further SEI formation and lithium consumption, leading to capacity and power drop. Severe pulverisation can lead to the electrical isolation of active material particles, rendering them inactive and reducing the overall capacity.

Thermal degradation is driven by the stress factor of temperature. Elevated temperatures accelerate chemical degradation rates via the Arrhenius effect, promoting rapid SEI growth, electrolyte decomposition, and transition metal dissolution. Conversely, low temperatures reduce ionic conductivity and slow down intercalation kinetics, significantly increasing the risk of lithium plating even at moderate charge rates. In large battery packs, uneven

temperature distribution causes heterogeneous current distribution, which accelerates local degradation in specific cells.

Aging Modes Degradation pathways are categorised by the operational state of the battery into cycling aging and calendar aging. Cycling aging occurs during charge and discharge operations and is driven by mechanical stresses from volume changes and the continuous reformation of SEI due to particle cracking and high current loads. Calendar aging occurs when the battery is at rest and is dominated by the passive growth of the SEI layer. This process is strongly influenced by the storage temperature and SOC, where higher SOC and temperatures accelerate capacity fade during storage.

1.1.2. Thermal Runaway

Thermal runaway (TR) represents the primary failure mechanism for lithium-ion batteries and is defined as a self-accelerating reaction where the rate of internal heat generation surpasses the battery's capacity to remove that heat [39]. This critical imbalance leads to a rapid and uncontrollable rise in temperature, often culminating in the generation of smoke, fire, and potential explosions. The phenomenon is essentially a chain reaction of decomposition events that escalates the internal energy of the cell. The process typically initiates with the decomposition of the SEI on the anode, which is followed by exothermic reactions between the anode and the electrolyte [40]. As the internal temperature continues to climb due to these initial reactions, the separator, which serves to electrically isolate the anode from the cathode, may melt or shrink. The failure of the separator allows the electrodes to come into direct contact, causing an internal short circuit that instantly generates a significant amount of heat and further accelerates the reaction kinetics. At high temperatures, the cathode material itself begins to decompose, a stage that is particularly critical because it releases oxygen. In addition, at $\approx 200\,^{\circ}\mathrm{C}$, the electrolyte evaporates releasing flammable gases [41]. Thus, the liberated oxygen reacts with these flammable gases and other combustible gases generated inside the cell, such as ethylene and propylene, thereby fueling combustion and leading to catastrophic failure [42].

The onset of thermal runaway is typically triggered by abusive conditions which can be classified into thermal, mechanical, and electrical categories. Thermal abuse involves exposure to excessive external heat that raises the internal temperature enough to initiate chemical instability. Mechanical abuse, such as crushing, penetration, or collision, physically damages the separator and electrodes, often leading directly to an internal short circuit. Similarly, electrical abuse conditions like overcharging, over-discharging, or external short circuits generate excessive Joule heating and induce chemical instability that can escalate into a runaway event.

The progression of thermal runaway can be characterised by specific temperature thresh-

olds that mark distinct stages of the failure. The process begins at the onset of self-heating, where the battery reaches a temperature between 70–100 °C [43, 44] depending on the specific conditions, that sustains exothermic reactions, primarily starting with SEI decomposition. As the temperature rises, the cell reaches a triggering temperature of about $\sim 130\,^{\circ}\mathrm{C}$ [45], where the separator typically collapses or melts, leading to an internal short circuit; at this point, the heat generation rate increases sharply as the anode and cathode react violently with the electrolyte. Finally, the cell reaches the maximum temperature in the range 600–1000 °C [43, 46], where the cathode decomposition is fully active, releasing oxygen and causing severe combustion of the electrolyte and generated gases. While internal short circuits are a common trigger, some research indicates that mechanisms such as "chemical crosstalk", where oxygen released from the cathode migrates to react with the anode, can also trigger thermal runaway even in the absence of a distinct short circuit event [47].

1.2. Battery Electric Vehicles (BEV)

Figure 1.5 depicts a representative configuration of power electronics and their integration with the energy storage and electric motor in a BEV. Electric motors in BEVs are responsible for converting electrical energy into mechanical work with high efficiency, typically ranging from 80% to 95% [22]. While electric machines exist in both direct and alternating current varieties, the alternating current (AC) machine is the standard for automotive applications. These motors operate on three-phase AC, utilising three voltage signals offset by 120°. When applied to the stator windings, these signals generate a rotary magnetic field that drives the rotor [48]. Among the various topologies available, the Permanent Magnet Synchronous Motor (PM-SM) and the Induction Motor (IM) have emerged as the industry standards for automotive traction. The PM-SM is widely favoured for its high power density and efficiency, achieved through the use of permanent magnets in the rotor which minimise heat generation [22]. However, this reliance on rare-earth materials can increase manufacturing costs. Conversely, the Induction Motor is valued for its robustness, simplicity, and lower production costs as it does not require permanent magnets. While IMs are highly reliable in harsh environments, they generally exhibit lower power density and require more complex control strategies to manage slip and optimise performance compared to synchronous machines.

Beyond the motor and battery, the power electronics module is a critical technology for vehicle efficiency and economy. This module typically includes two main categories of converters. The DC-DC converter is utilised to adjust the voltage level of the high-voltage DC current flowing from the batteries to the inverter; additionally, a low-voltage side is used to feed auxiliary loads such as lighting, the cabin Heating Ventilation and Air Conditioning (HVAC) system, infotainment, and safety systems. The second key component is the traction inverter (DC-AC), which converts the DC current from the high-voltage DC-DC converter into AC required by the motor. Furthermore, an AC-DC converter (On-Board Charger) can

be included to facilitate the charging process and support the onboard power supply.

Charging the vehicle from the external electrical grid requires a device to adapt the external current to the battery's specific voltage needs. Two technical solutions exist for this interface. The first option is the on-board charger: these units are integrated directly into the vehicle and are the standard for power levels between 3.6 and 22 kW. If the on-board charger is bi-directional, the current can flow in both directions, allowing energy transfer from the batteries pack to the grid. The second charging solution relies on off-board chargers: these are external units located at a charging column, typically utilised for fast-charging applications delivering 50 kW or more.

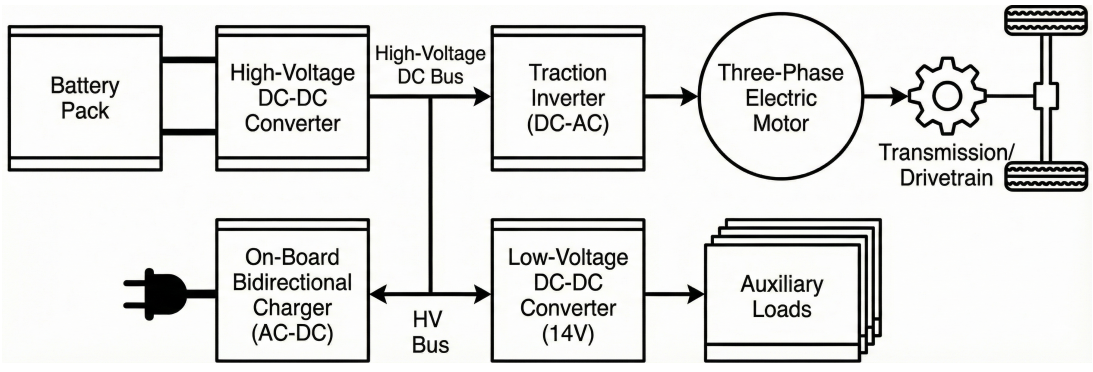


Figure 1.5: Scheme of powertrain configuration in a BEV.

Each component detailed in Figure 1.5 has its own optimal operating temperature which ensures maximum efficiency. In this context the thermal management system of the BEV has a fundamental role and needs to be designed to meet the desired operating conditions for each subsystem and prevent safety risks, such as thermal runaway.

1.2.1. Thermal Management System

The Thermal Management System (TMS) in BEVs presents higher complexity compared to Internal Combustion Engine Vehicles (ICEVs). In ICEVs, the thermal management system is usually composed of a single coolant loop (and therefore a single coolant pump) to cool the engine block, which is the main heat source, and the auxiliary components. The heat absorbed by the coolant is then transferred to the ambient air through a radiator, which is usually combined with a fan to enhance heat transfer. The cabin is warmed by exploiting the waste heat from the engine block, whilst an air conditioning system working with a refrigerant is usually employed for cooling.

In BEVs, the subsystems work within different operating temperature windows. As shown in Figure 1.6, the electric motor and the power electronics work typically in the range 70–110 °C, whereas the batteries should operate in the range 15–35 °C. The cabin air temperature depends on the specific driver and passengers but usually remains in the range 20–25 °C. Therefore, a single loop similar to the one used in ICEVs cannot ensure the correct

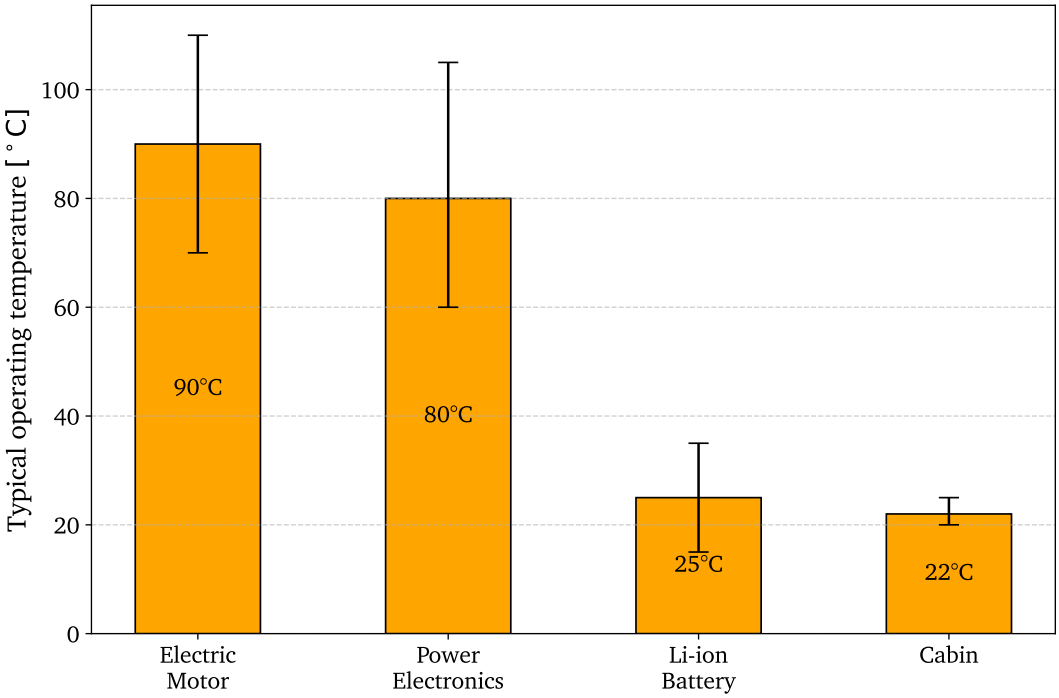


Figure 1.6: Indicative operating temperatures of BEV subsystems.

operating temperature window for each component. For this reason, the TMS in BEVs is more complex. The standard TMS is composed of two main loops: one for the motor and power electronics, and the other for the batteries. In addition, since the energy efficiency of the motor and batteries is much higher than that of an ICE ($\sim$ 90% vs $\sim$ 35%), the waste heat is lower and is insufficient to warm the cabin air. Consequently, separate systems are used for cabin air heating, such as Positive Temperature Coefficient (PTC) heat exchangers. Finally, as in ICEVs, cabin air cooling is usually ensured by an air conditioning unit.

Figure 1.7 schematises a common configuration for an Integrated Thermal Management System (ITMS), defined as the baseline configuration in [49]. This system integrates the loops of different subsystems to exploit the thermal power distribution more efficiently. The core is the Vapour Compression Cycle (VCC) which manages both cabin and battery thermal loads. Whilst the cabin benefits from direct refrigerant cooling, the battery pack is cooled indirectly through an intermediate coolant circuit containing a water-glycol mixture. In this cycle, after the refrigerant absorbs heat from the cabin environment, it is compressed by a scroll compressor and then releases heat to the external air through a condenser. Downstream of the condenser, the refrigerant path bifurcates into two parallel branches, each regulated by a specific expansion valve. The first branch directs refrigerant to the cabin evaporator for air conditioning. The second branch routes refrigerant to a chiller, where it absorbs heat from the battery coolant loop or, more generally, from the specific BTMS employed. To address low-temperature scenarios, such as cold starts after overnight soaking, the battery

loop incorporates an electric heater designed to raise the battery temperature to its target operating set-point. Separately, a distinct liquid cooling circuit is dedicated to the motor and power electronics; this loop utilises a pumped water-glycol mixture to extract waste heat from the components and transfer it to the external air via a radiator.

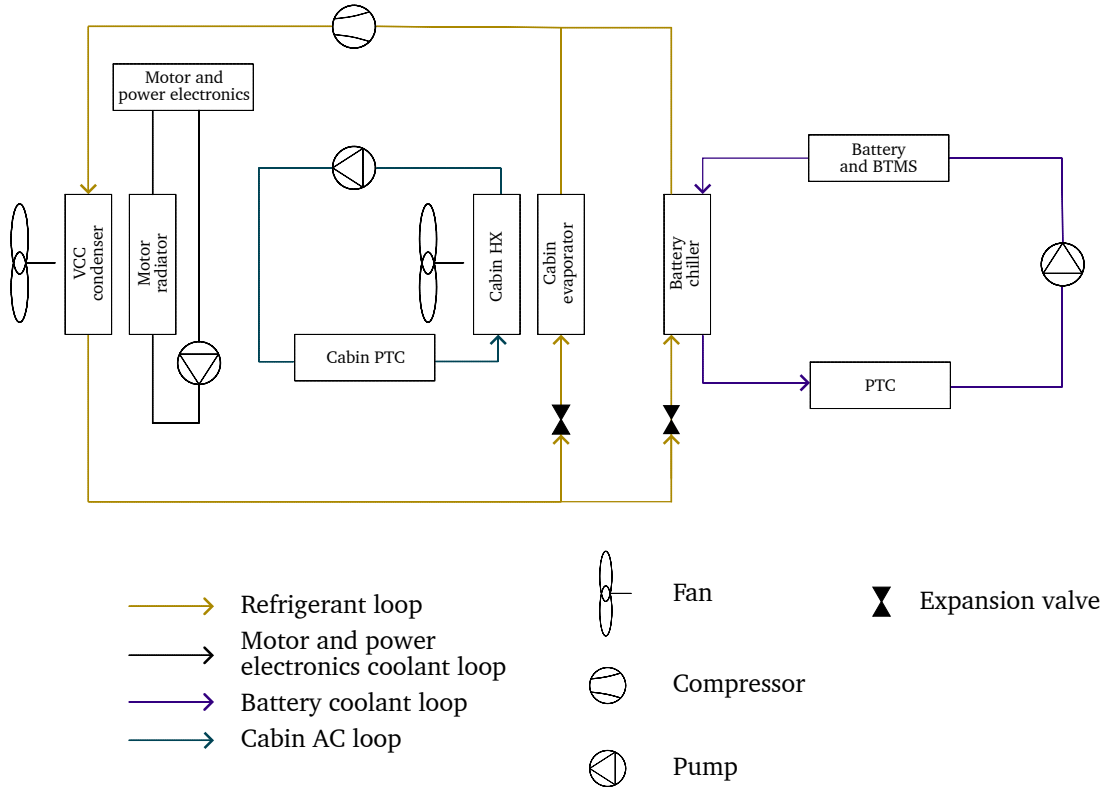
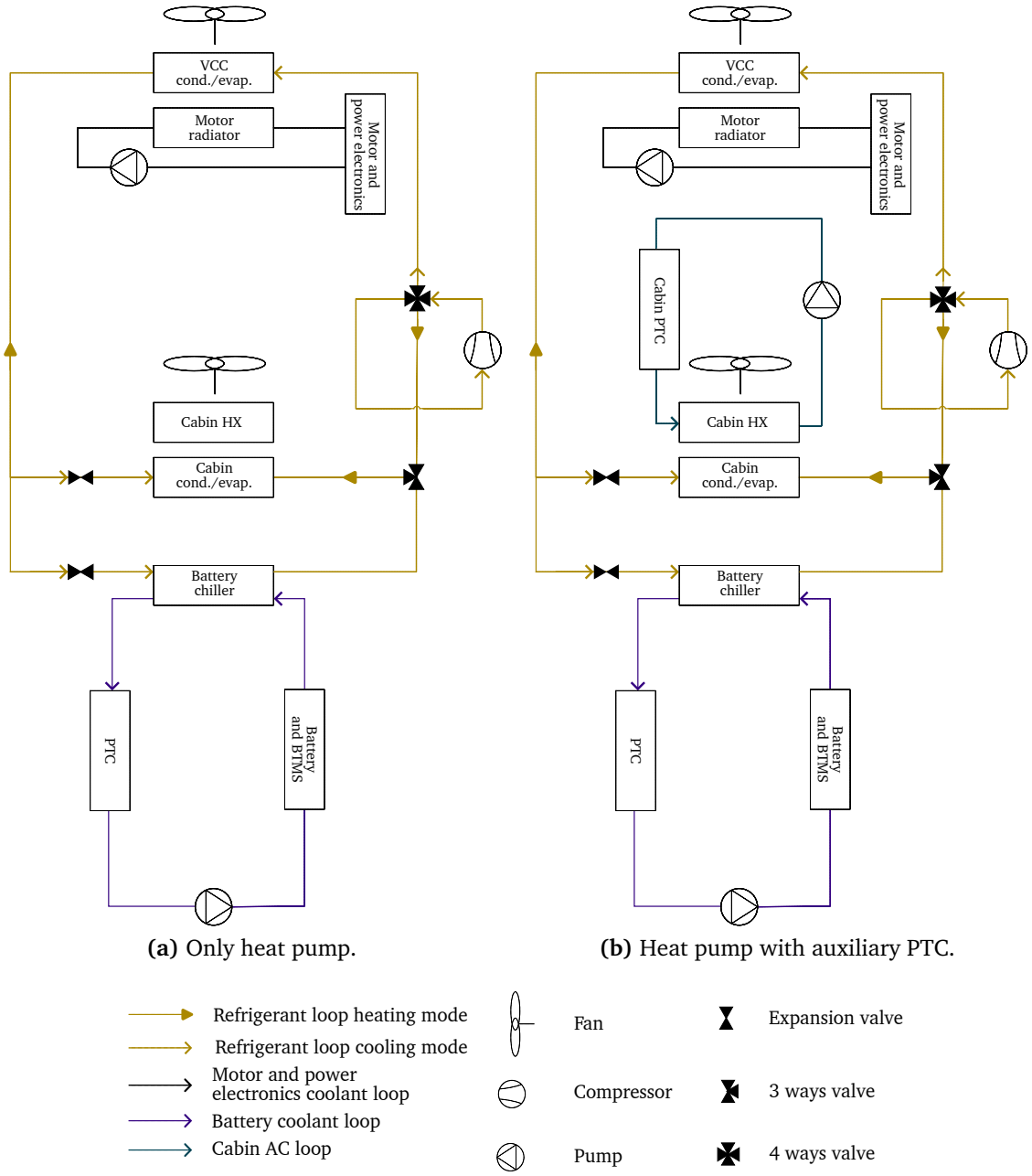


Figure 1.7: Base Integrated Thermal Management System (ITMS).

When the temperature difference between the cabin internal temperature and the set-point temperature is high, the use of a PTC for cabin heating can lead to a significant decrease in the vehicle's driving range. Therefore, an improvement can be achieved by integrating a heat pump which substitutes the PTC with a more efficient cycle, since heat pumps operate with a COP > 1. This system is schematised in Figure 1.8a, where the PTC cabin heater loop is replaced by the heat pump. To use the heat pump in both cooling and heating modes, a four-way valve is necessary. In cooling mode, the refrigerant follows the same loop as the base ITMS. In heating mode, the flow direction is reversed so that the superheated vapour exiting the compressor flows to the cabin heat exchanger, which in this case acts as a condenser, whilst the external radiator acts as an evaporator. During heating mode, the three-way valve closes the path to the battery chiller. However, when the external temperature is $\lesssim 0\,^{\circ}\mathrm{C}$, the heat pump may not be able to ensure the set-point temperature in the cabin. In this context, a possible solution is shown in Figure 1.8b, where the PTC loop is restored to assist the heat pump under extreme cold conditions. According to the work by Shelly *et al.* [49], the TMS with a heat pump and PTC can increase the range of the drive cycle by $\sim 5\%$ with respect to

the base system at −10 °C.



(a) Only heat pump.

(b) Heat pump with auxiliary PTC.

Figure 1.8: ITMS configurations with the integration of a heat pump system.

A further step to optimise the energy flow in the TMS is the recovery of waste heat from the motor and power electronics loops. This system is schematised in Figure 1.9. The main difference compared to the system with a heat pump and PTC is the motor and power electronics cooling loop. After the coolant passes through these components, it reaches a four-way valve which can either connect the battery coolant loop in series, when the batteries need to be heated, or bypass it. In both situations, as the motor coolant returns, and before going towards the radiator, it encounters another four-way valve which can join the heated

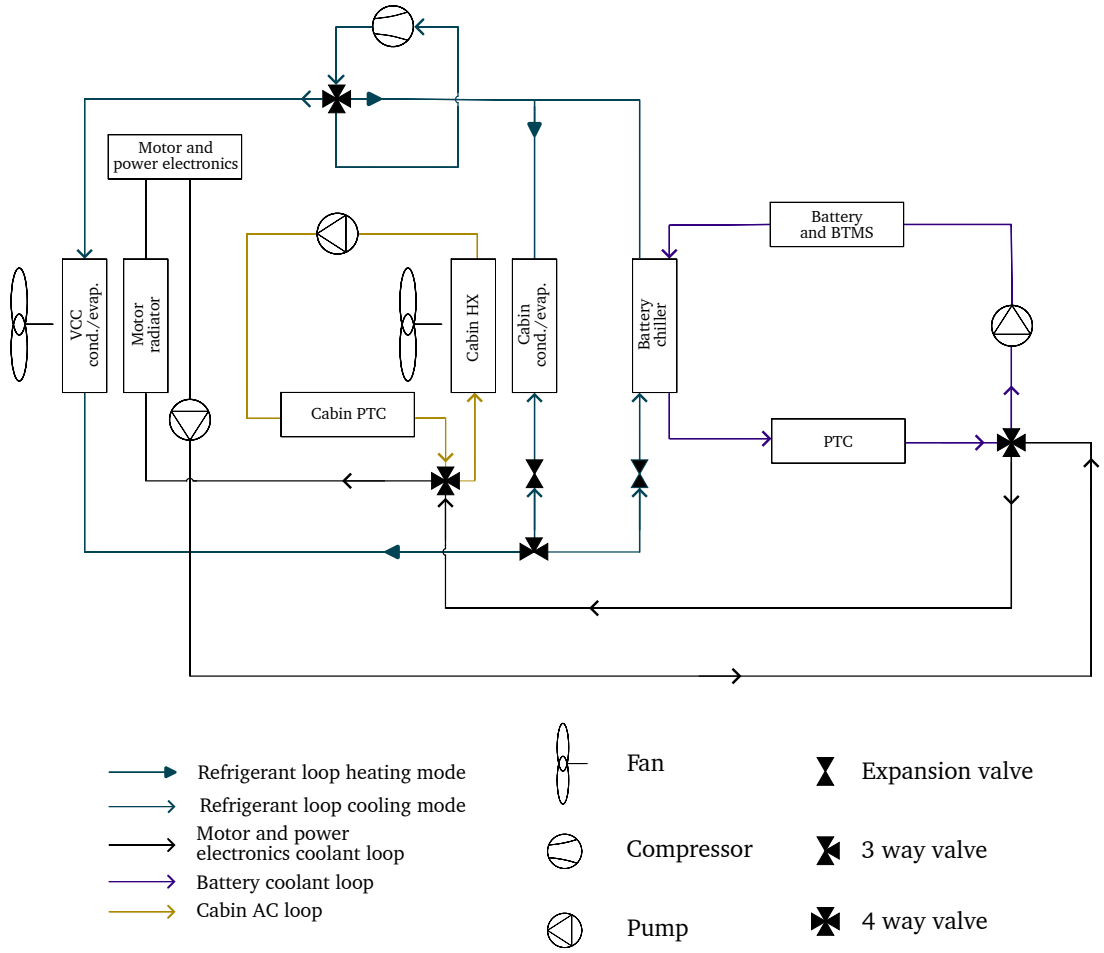


Figure 1.9: ITMS with heat pump, PTC and waste heat recovery.

coolant with the PTC loop. In this way, the heat available in the coolant can be used to heat the cabin. Furthermore, the four-way valve can be regulated between 0%–100% to manage how much waste heat is used for the cabin. In this type of configuration, the control system for cabin heating prioritises the use of waste heat. If this is insufficient, the heat pump is activated, and when the heat pump is also insufficient, the PTC is activated, which is the subsystem with the lowest efficiency. In cooling mode, the system works identically to the configuration in Figure 1.8b. At extreme cold conditions, this system surpasses the previous one, extending the driving range of the car by 13.5% at −10 °C [49].

Another step to optimise energy management could be to recover waste heat from the batteries. However, the batteries are usually between 15–35 °C, meaning there is a very low temperature difference with the cabin environment. In cold conditions, the temperature difference between the batteries and fresh air is greater, and the waste heat from the batteries can be used to heat the fresh air used to defog the windshield. The drawback is that the system becomes more complex, requiring an additional heat exchanger and valves, whilst the gain in energy saving does not justify the added complexity.

Another type of ITMS of interest is direct refrigerant cooling. Unlike indirect systems that

require a secondary loop containing a water-glycol mixture and a chiller to transfer heat, direct cooling routes the refrigerant directly into the cooling plates integrated within the battery pack. Figure 1.10 shows the layout of the ITMS with direct refrigerant cooling. The system relies on the vapour-liquid phase change of the refrigerant. After throttling and depressurising via an expansion valve, the two-phase refrigerant enters the cooling plate where it absorbs heat generated by the battery cells through evaporation. This phase change heat transfer possesses a significantly higher heat transfer capacity compared to the single-phase convection heat transfer utilised in liquid cooling systems, as the refrigerant temperature remains constant during phase change. To ensure the correct operating temperature in extreme cold conditions, the PTC is still required, and in the scheme it is directly integrated with the battery pack. By eliminating intermediate components like the coolant pump and the chiller, this architecture reduces the complexity and cost of the vehicle's thermal management system. However, whilst the technology theoretically offers superior temperature suppression, its practical application faces challenges regarding temperature uniformity [50]. The cooling performance is highly dependent on the state of the refrigerant. When the vapour quality is between 0 and 1, heat removal is more effective, whereas, when transition to a single-phase gas is completed, the heat transfer capability decreases. This potentially leads to significant local temperature spikes and uneven thermal distribution across the battery module.

Furthermore, since the cooling plate often shares the refrigeration cycle with the cabin air conditioning, the evaporation pressure and temperature are constrained by the requirements of the passenger compartment, limiting the system's flexibility. To address this problem, Guo *et al.* [51] proposed an ITMS with a direct refrigerant system featuring two stages of expansion for the battery loop. This makes it possible to differentiate the evaporation temperature of the refrigerant when passing through the batteries versus the cabin evaporator. The system is used for both cooling and heating. Figure 1.11a depicts the system in cooling mode. The grey dashed branches represent branches that exist in the system but are not used for this specific mode. The working principle is similar to that in Figure 1.10, but with the addition of an extra expansion valve after the batteries. In this way, it is possible to stage the expansion of the refrigerant going into the batteries to maintain a higher refrigerant temperature before it enters the battery plate. Afterwards, it undergoes another expansion to complete the cycle and meet the requirements for entering the compressor along with the refrigerant coming from the cabin loop. In fact, the refrigerant temperature in the cabin evaporator should be lower, since the target temperature is different and the heat transfer coefficient of the cabin heat exchanger differs from that of the plate integrated into the battery pack. Figure 1.11b schematises the system in heating mode. In this case, the superheated refrigerant is split between the battery and the cabin. After condensing and re-joining the two streams, it flows to the external evaporator, where it can also exploit waste heat from the motor, before restarting the cycle from the compressor. Finally, in the

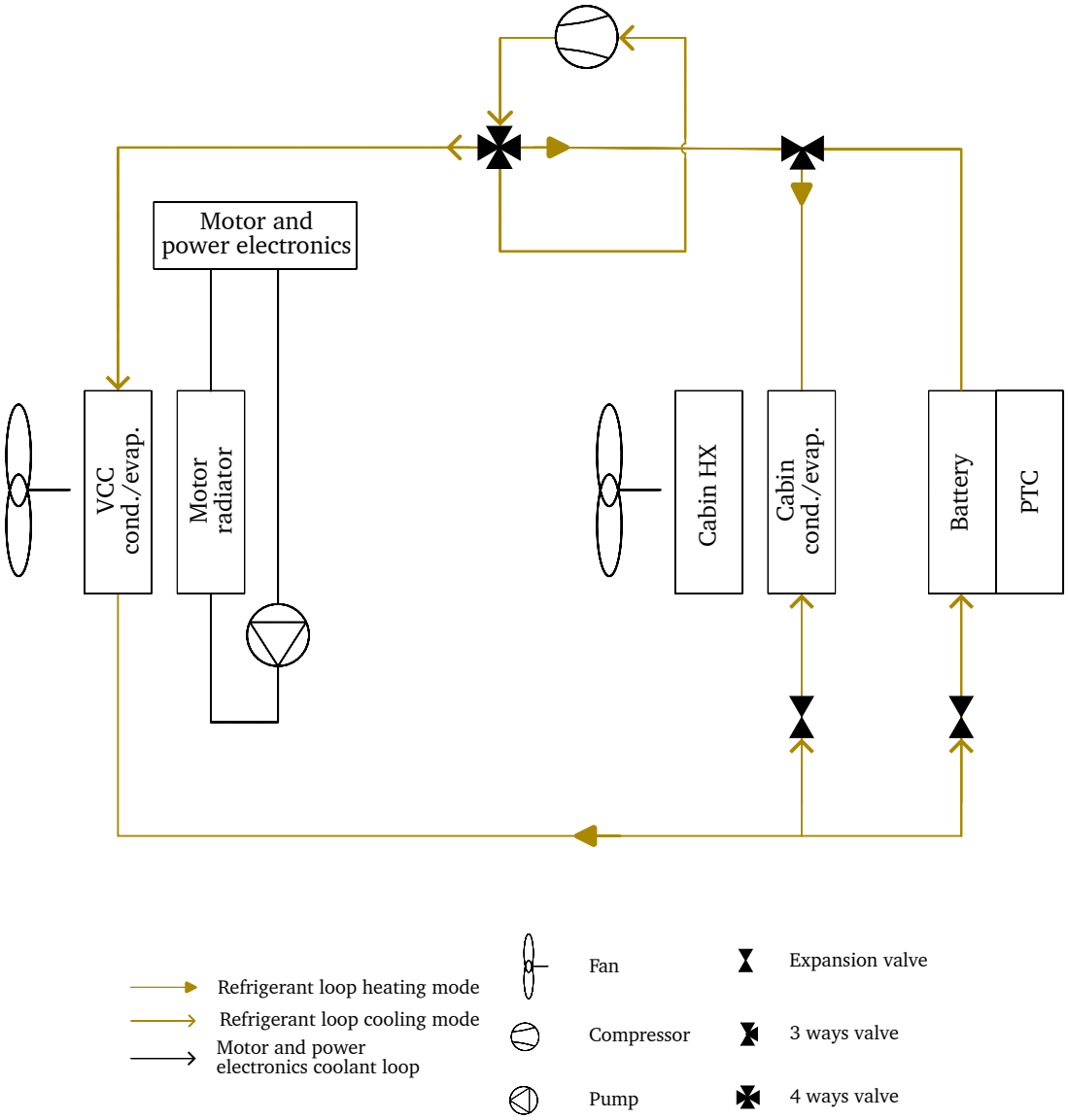
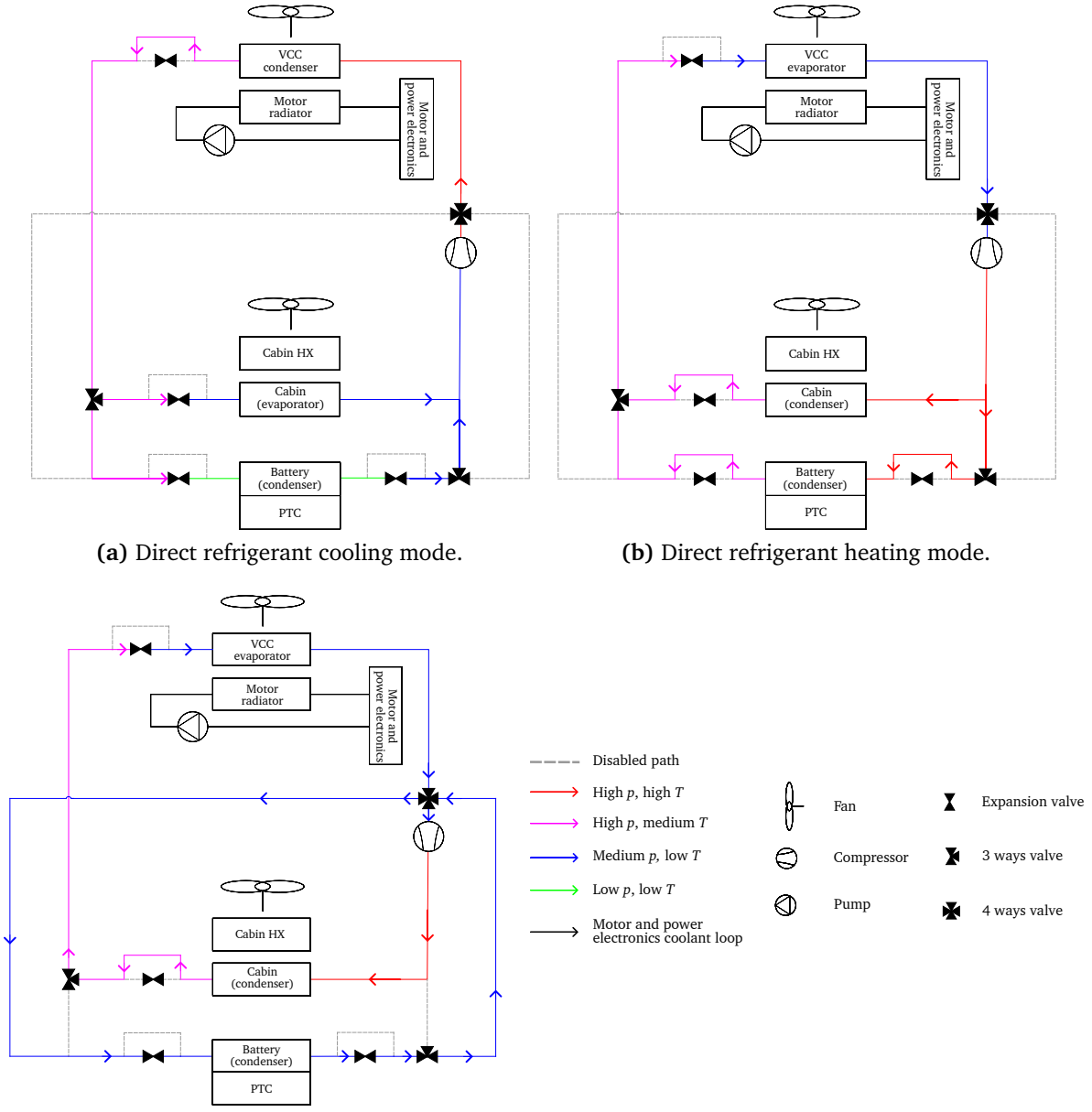


Figure 1.10: Basic TMS with direct refrigerant cooling.

same work a mixed mode is proposed, where the cabin is heated and the battery cooled (Figure 1.11c). In this case, the refrigerant exiting the compressor is not split but flows only towards the cabin condenser. After passing the external evaporator, instead of restarting the cycle, the refrigerant flows at low pressure and temperature towards the battery pack, where it is further heated before entering the compressor. This last configuration is used in non-extreme cold conditions when the cabin needs to be heated, recovering waste heat from the batteries. The authors of the mentioned work [51] simulated the proposed system under the standardised New European Driving Cycle (NEDC), confirming the effectiveness of direct refrigerant cooling.

However, the coolant loop is still preferred in the industry because the technology is more mature and the direct refrigerant system is highly sensitive to specific conditions. In fact, the phase change temperature does not always meet the specific requirements for both the battery

and the cabin. According to the review by Zhu *et al.* [52], the ITMS is identified as the future trend for Battery Electric Vehicles, offering approximately 20% greater energy savings than partially integrated systems by maximizing waste heat recovery. However, the technology is currently limited by a reliance on independent feedback controls for subsystems rather than a global, intelligent control strategy for the entire vehicle. Future perspectives therefore focus on developing modularised system structures and establishing a comprehensive evaluation system to optimise energy distribution and reliability at the vehicle level.



(a) Direct refrigerant cooling mode.

(b) Direct refrigerant heating mode.

(c) Direct refrigerant heating mode for the cabin and cooling mode for the battery.

Figure 1.11: Direct refrigerant ITMS.

Among the subsystems comprising the TMS of a BEV, the BTMS is the most sensitive to

temperature management, as it possesses a limited optimal temperature range and requires a strict temperature uniformity. The performance of the BTMS has a significant impact on energy consumption and battery durability. For this reason, most research regarding the thermal design of BEVs focuses on the BTMS.

1.2.2. Battery Thermal System (BTMS)

The primary function of the BTMS is to maintain the battery pack within its optimal operating temperature range (15–35 °C), whilst ensuring uniform temperature distribution among individual cells ($\Delta T \leq 5\,^{\circ}\mathrm{C}$). This prevents significant mismatches in operating conditions between the cells in the modules, which can reduce performance and accelerate degradation. Effective thermal management is critical because Lithium-ion batteries are highly sensitive to temperature; high temperatures accelerate degradation and risk thermal runaway, while low temperatures increase internal impedance and cause lithium plating. Consequently, the BTMS must effectively remove heat during high rate charging and discharging, and provide heating during cold conditions to preserve the safety, longevity, and electrochemical performance of the battery. Based on the physical properties of the cooling medium, BTMS is generally categorised into three main types: air cooling, liquid cooling, and Phase Change Material (PCM) [53].

The techniques mentioned above are the most common thermal management methods for BTMS. Furthermore, other types of cooling are being investigated, such as heat pipes, thermoelectric devices, and hybrid methods.

1.2.2.1. Air based BTMS

Air cooling systems exploit airflow to remove heat from the batteries and are characterised by their simple structure, low cost, low weight, and ease of maintenance. Despite these advantages, air cooling cannot be employed as the sole thermal management strategy for high performance vehicles: the inherently low thermal conductivity and specific heat capacity of air result in a limited convective heat transfer coefficient.

Depending on the mechanism driving the flow, air cooling systems are classified into *natural* and *forced* convection. Natural convection relies on spontaneous buoyancy airflow driven by temperature differences between the batteries and the surroundings. This passive system is the simplest and does not require specific components to operate. However, this mode of cooling is not suitable for high power applications; for this reason, it is not usually employed as the main thermal management strategy in BEVs.

Forced convection systems force airflow into the modules using fans. This can be passive, using ambient or cabin air, or active, using pre-conditioned air from the vehicle's HVAC system. Passive forced convection systems have a limited operating range because their thermal management performance is highly dependent on the ambient air temperature [54];

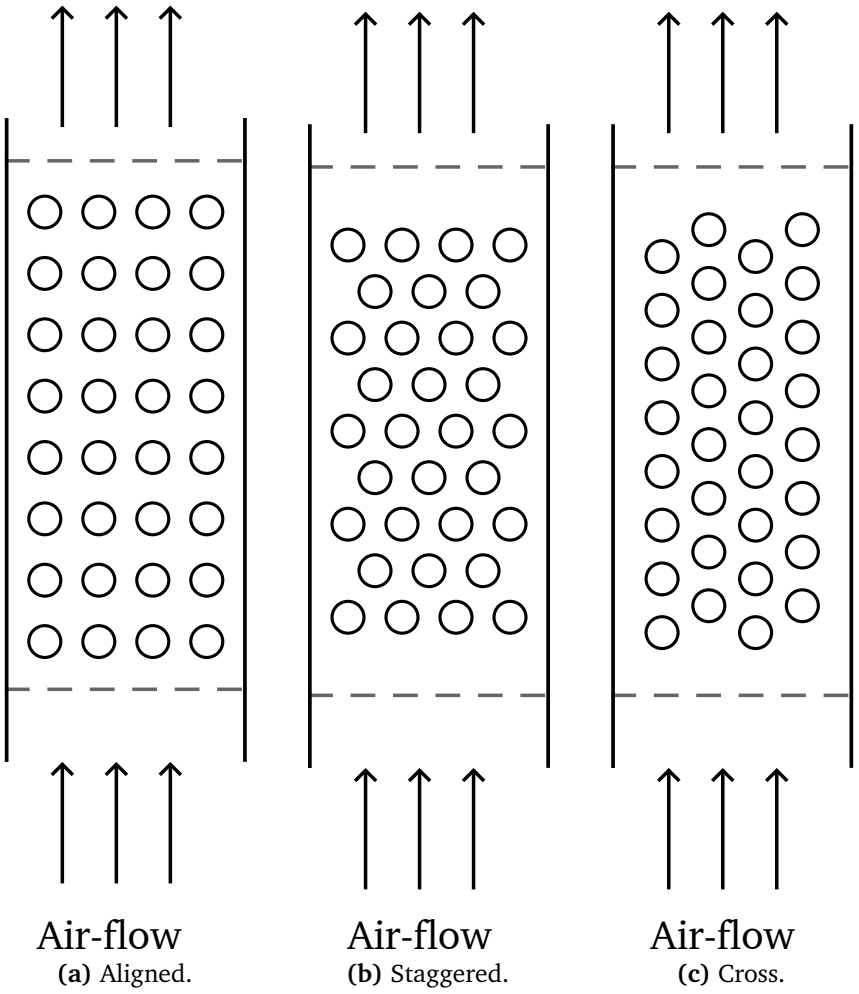
therefore, active systems are generally preferred. Different studies on forced convection systems focus on the airflow path between the batteries to optimise the temperature range and uniformity by designing different structures of cell arrangements. Fan *et al.* [55] performed an experimental investigation on different cylindrical cell arrangements, depicted in Figure 1.12, for a forced air cooled module. The results indicated that the aligned arrangement offered the best cooling effectiveness. Additionally, the second row of cells (in the direction of the airflow) exhibited the lowest temperature in all the cases, as the air temperature remains near the inlet value and the flow accelerates after passing the first row, improving convective heat transfer. Other studies focus on duct configurations [56]. For example, the U-type configuration (Figure 1.13a) leads to better temperature uniformity between the cells because the inlet and the outlet are on the same side. The work by Wang *et al.* [57] compared different parallel configurations (including U-type) varying the positions for duct inlets and outlets by performing Computational Fluid Dynamic (CFD) simulations. Their results indicated that the configuration depicted in Figure 1.13b is the best among those tested, as having ducts parallel to the channels between the cells leads to higher airflow velocity and ensures good temperature uniformity. In a parallel configuration, tapering the ducts can balance airflow fluctuations among the gaps in the modules. Sun *et al.* [58] investigated a Z-type tapered configuration where the inlet and outlet are on opposite sides, both horizontally and vertically, such that their arrangement forms a diagonal (Figure 1.13c). They developed a DOE procedure to predict the performance of the cooling system under different geometric parameters and flow conditions.

When the modules comprise cylindrical batteries, the axial configuration can be a good solution to minimise the system volume while maintaining thermal performance. In this configuration, the airflow is forced axially through the gap between the cylindrical cells, as shown in Figure 1.14. According to Yang *et al.* [59], increasing the radial spacing between the cells improves temperature uniformity, while the average temperature increases slightly. In addition, the power required to overcome pressure losses is lower when the gap is increased. In this context, it is important to find the best trade-off between cell spacing and the volume coefficient of the modules.

Since air cooling systems present a low heat transfer coefficient, some studies propose finned structures to improve heat removal. However, these structures significantly increase the weight of the battery pack, resulting in a heavier system compared to liquid cooling solutions, with the latter offering a higher heat transfer coefficient and better thermal uniformity.

1.2.2.2. Liquid based BTMS

Liquid cooling utilises fluids with higher thermal conductivity and specific heat capacity with respect to air, making it more suitable for large-scale battery packs operating at high

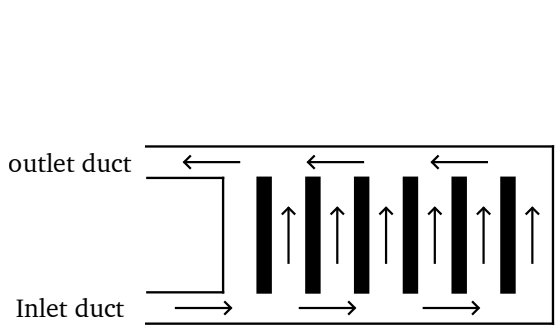


(a) Aligned. **(b)** Staggered. **(c)** Cross.

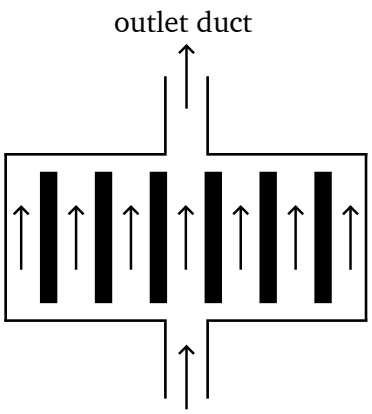
Figure 1.12: Cylindrical cells arrangements investigated in [55].

charge and discharge rates. Consequently, liquid systems are the most widely used solution for high-performance BEVs. Table 1.1 details the battery capacity and cooling methods of several commercial electric vehicles. In these vehicles, both liquid and air cooling methods are employed, though liquid cooling is predominant. To the best of the author's knowledge, the electric vehicle with the largest battery capacity employing an air-based BTMS is the new generation Nissan Leaf, with a capacity of 62 kWh. Therefore, this value can be considered the current threshold between air cooling and liquid cooling applications. Indeed, all vehicles with larger capacities utilise liquid cooling for their battery packs.

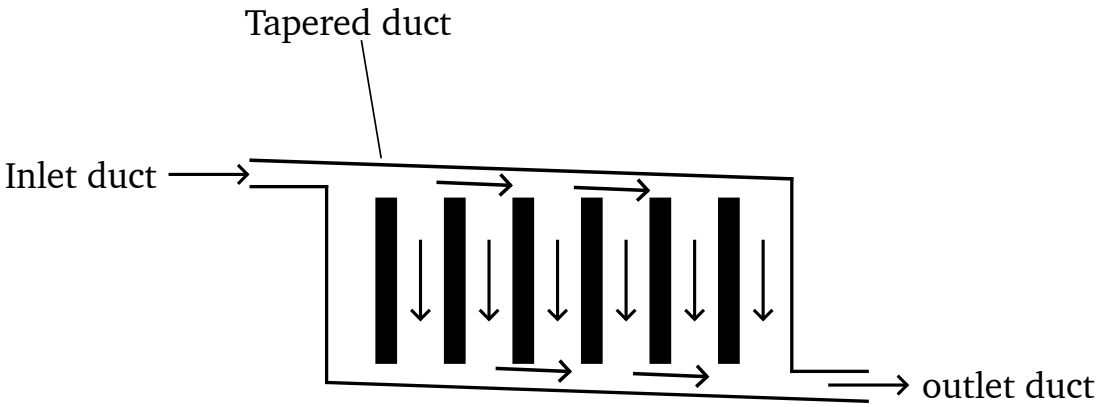
Based on the contact interface between the liquid and the batteries, liquid cooling systems are categorised into direct and indirect liquid cooling [61]. In direct cooling mode, the liquid is in contact with the external surface of the batteries, minimising thermal resistance between the coolant and the cells. Generally, the convective heat transfer of direct cooling is one order of magnitude greater than that of traditional air cooling systems, which also enhances temperature uniformity within the modules. The cooling medium employed for direct liquid



(a) U-type configuration.



(b) Parallel longitudinal configuration.



(c) Z-type tapered configuration.

Figure 1.13: Ducts configurations investigated in [56, 57] and [58].

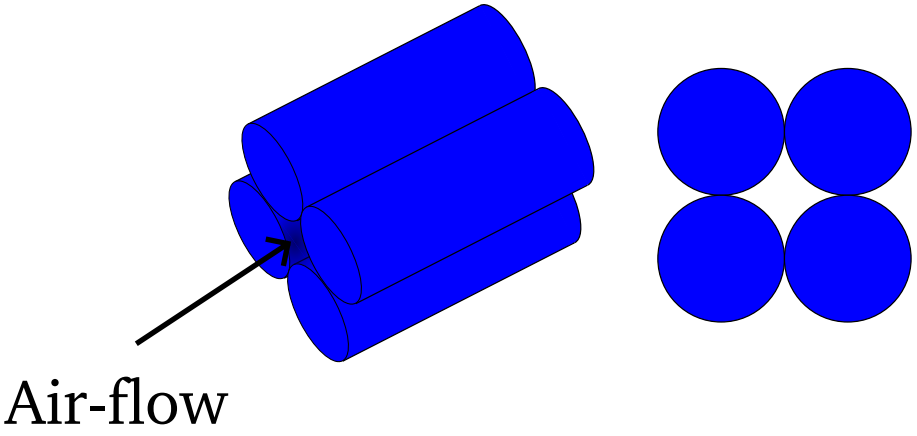


Figure 1.14: Axial air cooling.

Table 1.1: Thermal management specifications of famous electric vehicles; data from [60].

Model	Battery size [kWh]	Cooling method
Rivian R1S	135.0	Indirect liquid cooling
Lucid Air Dream Edition	118.0	Indirect liquid cooling
Mercedes-Benz EQS 580	107.8	Indirect liquid cooling
Tesla Model S Plaid	100.0	Indirect liquid cooling
Audi e-tron S Sportback	95.0	Indirect liquid cooling
Porsche Taycan Turbo S	93.4	Indirect liquid cooling
BMW i4 eDrive40	84.3	Indirect liquid cooling
BYD Seal	82.5	Indirect liquid cooling
Tesla Model 3 Long Range	82.1	Indirect liquid cooling
Hyundai Ioniq 5	77.4	Indirect liquid cooling
Kia EV6	77.4	Indirect liquid cooling
Ford Mustang Mach-E	75.7	Indirect liquid cooling
Toyota bZ4X	72.8	Indirect liquid cooling
Chevrolet Bolt EV	65.0	Indirect liquid cooling
Nissan Leaf (2019)	62.0	Forced air convection
Nissan Leaf (1^{st} gen.)	40.0	Forced air convection
Kia Soul EV	27.0	Forced air convection
Renault Zoe R240	25.92	Forced air convection

cooling requires excellent chemical-physical properties, including high thermal conductivity and heat capacity, and low viscosity. Furthermore, it must be electrically insulating, as it is in direct contact with the cells, posing a risk of short circuits. Additionally, it must be non-flammable and non-toxic to meet environmental safety requirements. Suitable media for direct cooling include silicone oils, hydrocarbon oils, and fluorinated hydrocarbons; water-glycol based coolants are not recommended due to their electrical conductivity. Chen *et al.* [62] numerically investigated different cooling methods: air cooling, direct liquid cooling, indirect liquid cooling, and fin cooling. They concluded that fin cooling adds the most weight, while air cooling consumes the most parasitic power; direct liquid cooling yielded the best results.

Direct cooling can be either single-phase or two-phase, where the latent heat of vaporisation of the coolant is exploited to maintain a constant temperature.

Direct liquid cooling is a promising solution that outperforms other BTMS [63], as the cells are immersed in a highly thermally conductive fluid. Moreover, it can potentially contain thermal runaway events, thereby improving safety in BEVs. Currently, direct liquid cooling is not common in commercial BEVs due to added complexity, high fluid and maintenance

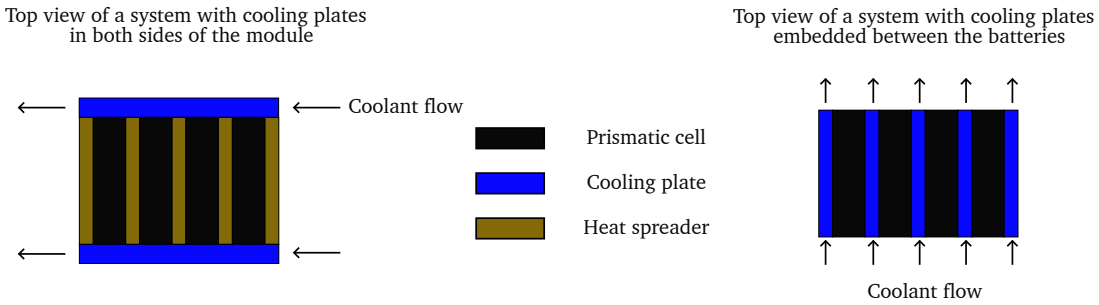
costs, compatibility issues with module materials, increased pumping power requirements, and the additional weight imposed on the ESS [64].

Indirect liquid cooling is the most common method and can be implemented in various ways: cooling plates, discrete tubes snaking through the cells, or microchannels positioned near the cells. Cooling plates are the most mature solution; they also function as structural components and can be manufactured in different shapes and positioned in various configurations, such as on one side only, on both sides forming a sandwich, or embedded between the cells. These solutions are depicted in Figure 1.15. When cooling plates are positioned on the side of the modules, a high-conductivity material (labelled as heat spreader in the figure) can be used to reduce thermal resistance between the batteries and the plates, although this increases the module weight. With the embedded solution this is not necessary, as the plates are in contact with the prismatic cell surface, resulting in very effective heat transfer. However, system complexity increases, and installation costs are higher. Panchal *et al.* [65] experimentally investigated the cooling performance of a cooling plate in direct contact with a prismatic cell and subsequently validated a CFD model. They discussed the influence of the discharge rate and ambient temperature. Other studies focus on investigating different channel geometries to enhance heat transfer. Huo *et al.* used a numerical approach to investigate the effect of flow rate, number of channels, flow direction, and inlet temperature on the thermal performance of parallel mini-channels cooling plate. Their results indicated that increasing the number of channels decreases the battery temperature; increasing the flow rate is also beneficial, but exceeding the optimal threshold results in a decrease in performance. Jarret and Kim [66] conducted a parametric optimisation to reduce temperature, maximise temperature uniformity, and decrease pressure losses in cooling plates with a serpentine channel. They conducted multiple CFD simulations varying different geometric parameters. The results showed that the configuration with the maximum channel width yields the minimum temperature and pressure losses, while the configuration that provides the best temperature uniformity is the one where the channel is narrowed at the inlet and widens towards the outlet, balancing the heat transfer area and the fluid-solid temperature gradient. Wang *et al.* [67] conducted a topology optimisation for a battery cooling plate considering both laminar and turbulent flow. From the optimisation, they obtained leaf-like structures that minimised the average temperature and pressure drop. The solution with turbulent flow shows better results, while the solutions with laminar flow were found to be inadequate to fully meet cooling requirements. Gao *et al.* [68] numerically investigated a cooling plate featuring gradually varied circular notched fins to increase turbulence, leading to better cooling performance. They found that increasing the diameter of the notched fins decreases the battery temperature.

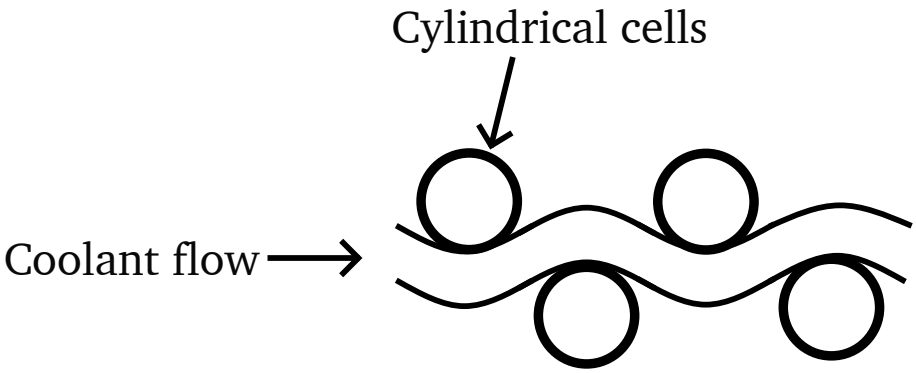
Due to the flat shape of the plates, cooling plates are widely used with prismatic cells to maximise the heat transfer surface. For cylindrical cells, a solution to improve the heat

transfer coefficient is the use of discrete tubes snaking between the cells (Figure 1.15b), as in the BTMS of the Tesla Model S. Zhang *et al.* [69] demonstrated that this configuration can achieve the same performance as a cooling plate but with less fluid volume and a shorter flow path. However, these types of systems are more complex to install, and maintenance costs are higher. A similar approach is proposed in the work by Murphy and Akrami [70], embedding cylindrical cells within a cooling matrix; they compared three different flow configurations: the first with vertical flow parallel to the cell axis (Figure 1.16), the second with horizontal flow perpendicular to the cell axis, and the third being a combination of the first two, with both vertical and horizontal channels. The vertical flow shows better thermal performance with respect to the horizontal flow configuration, while the third is superior to both of the first two solutions; however, the third configuration requires more components and a larger volume of fluid. Falcone *et al.* [71] proposed a BTMS in which cylindrical cells are embedded within a high-conductivity plastic matrix, featuring horizontal wavy channels extracted between the cells. They analysed the proposed system using a CFD model validated against experimental results. The model was employed to compare the performance of air and water as working fluids and to perform an optimisation of the cell arrangement and channel geometry parameters. They found that when air is the working fluid, the optimal strategy is to increase the channel size, whereas with water, it is preferable to reduce it. Additionally, the results showed that using water ensures lower cell temperatures compared to air. The use of a high-conductivity plastic matrix represents a favourable trade-off between thermal performance and weight, as plastic is lighter than common metal materials used as matrix.

In the context of liquid cooling, there are various studies regarding the Heat Transfer Fluid (HTF). The most common HTF in indirect liquid cooling is water or a mixture of water-glycol, due to its thermal properties, chemical stability and low cost. However, various researchers have investigated alternatives to overcome the limitations of water regarding thermal conductivity and heat capacity. Huo and Rao [72] developed a Lattice-Boltzmann model to study the effects of adding Al_2O_3 nanoparticles to water, exploiting the higher thermal conductivity of aluminium. They found that with a volume fraction of 0.04, the average battery temperature decreases by 7% compared to water alone. Yang *et al.* [73] conducted CFD simulations to investigate the performance of aluminium-based liquid metal. Liquid metal offers several advantages compared to water; under the same flow conditions and with lower pumping power, lower temperatures and higher temperature uniformity are achieved, thanks to the higher thermal conductivity. Another advantage is the reduced risk of leakage, as the liquid metal in question possesses a surface tension ten times higher than that of water. Furthermore, by employing an electromagnetic pump, the liquid metal flow can be easily reciprocated. However, since the density is six times higher compared to that of water, liquid metal adds significantly more weight to the system.



(a) Top view of liquid cooling systems with prismatic batteries.



(b) Discrete tubes cooling for cylindrical batteries.

Figure 1.15: Indirect liquid cooling solutions.

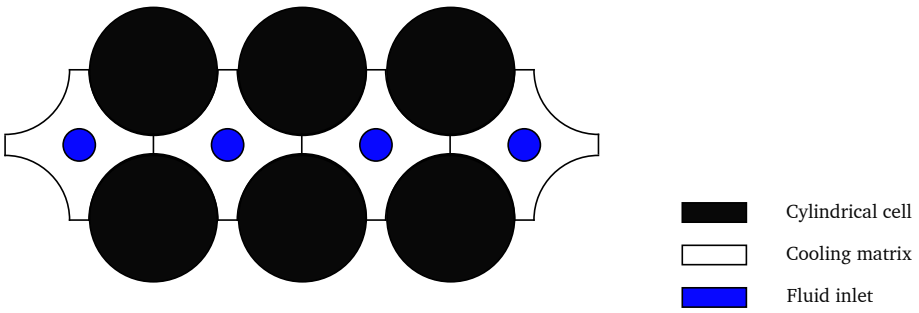


Figure 1.16: Top view of a cooling matrix for cylindrical cells with vertical flow.

As previously discussed regarding the overall TMS of BEVs, another solution is the use of direct refrigerant cooling, which exploits the latent heat of vaporisation to cool the cells at a constant temperature. The same principle is employed in immersed boiling methods, as described in the work by Al-Zareer *et al.* [74], who proposed a direct liquid cooling method where the batteries are partially immersed in a pool of boiling liquid. They used liquid ammonia as the HTF, which boils at $\sim$ 20–25 °C; when the liquid boils, the vapour cools the fraction of cells not immersed in the liquid via natural convection. By means of Finite Element Analysis (FEA), they found that increasing the saturation pressure of the liquid ammonia increases the maximum battery temperature, but also improves temperature uniformity. Furthermore, covering 5% of the cells with liquid ammonia at 10 bar can maintain the internal temperature difference below 3 °C during 300 s at a 2.5C discharge rate.

1.2.2.3. Phase Change Material (PCM)

PCM refers to a solid material which absorbs or releases heat according to the phase change process. The advantage is that it has a low melting point (in the optimal temperature range of batteries) and, thanks to its latent heat, it can store a noteworthy amount of heat at constant temperature during the melting process. One of the biggest benefits is that PCMs can maintain a proper battery temperature and temperature uniformity without the need of extra power consumption [75]. One of the main challenges of pure PCMs is the low thermal conductivity of the most common materials. Generally organic materials such as Paraffins, fatty acids, and polyethylene glycol are employed, which offer significant latent heat capacity, chemical inertness, and adjustable melting temperatures, all of which are critical for preventing thermal runaway during battery cycling. However, their drawback is their low thermal conductivity. Other challenges regard structural integrity, weight and volume changes during the phase change cycles and leakage risk by the melted material. In addition after the melting process is completed, it is necessary to restore the material by inducing solidification, therefore it is necessary to decrease the temperature of the melted PCM to the solidification temperature.

To address these challenges, Composite Phase Change Material (CPCM) can be employed. Wu *et al.* [76] added a copper mesh to a paraffin PCM with expanded graphite, where the paraffin exploits his high latent heat to store heat generated within the batteries, the expanded graphite adsorbs the liquid paraffin to prevent leakage and to transfer the heat to the copper more effectively. The copper mesh is finned, so that, by including a forced airflow, the heat dissipates more effectively. Oya *et al.* [77], to enhance the thermal conductivity, added expanded graphite and nickel nanoparticles to a PCM composed of erythritol. From their experiments they found that the CPCM can achieve a 640% higher thermal conductivity compared to pure PCM. Lv *et al.* [78] conducted experimental tests on a battery module with a CPCM, by adding silica nanoparticles to an organic PCM to decrease the leakage risk

and also the volume changes.

Usually, PCM or CPCM systems alone, are not enough to manage the temperature of high energy density battery packs , therefore most of the applications of these solution are in combination with other system such as air cooling, liquid cooling or heat pipes [79]. Table 1.2 reports and compares the characteristics of the main BTMS strategies. The data in the table comes from [80] which reviewed the BTMS for new generation vehicles. The air cooling is the most convenient in the context of flexibility, cost and durability. However, the energy density and poor temperature uniformity make this strategy not suitable for most BEV applications. PCM can ensure optimal temperature distribution with a moderate cost and power consumption, but also in this case the energy density is low and there are challenges due to weight and leakage. Finally, liquid cooling is the most suitable for high energy density applications, but it is the most expensive and requires also more maintenance.

Table 1.2: Characteristics of air cooling, liquid cooling and PCM for BTMS; data from [53].

	Air forced	Liquid	PCM
Life	$\geq$ 20 years	3-5 years	$\geq$ 20 years
Ease of use	Easy	Difficult	Easy
Integration	Easy	Difficult	Easy
Energy density	Low	High	Low
Maintenance	Easy	Difficult	Easy
Temperature distribution	Uneven	Even	Even
Efficiency	Low	High	High
Temperature drop in cell	Small	Large	Large
Annual cost	Low	High	Low
First cost	Low	High	Moderate

1.2.2.4. Heat pipes (HPs)

HPs are passive thermal management systems that exploit pressure gradients and capillarity effects to ensure fluid motion. Figure 1.17 schematises the fundamental principle of a standard sintered heat pipe. The interior of the heat pipe is evacuated of air and contains a small amount of working fluid, commonly water. Consequently, the internal environment is at saturation conditions, promoting the phase change of the working fluid. The heat pipe can be divided into three sections: the evaporator section at one end, the condenser section at the other end, and the adiabatic section in between. In the evaporator section, the working fluid absorbs heat and evaporates, creating a local high pressure zone due to expansion during the phase change. Driven by the pressure difference, the vapour travels towards the other end, where it condenses, reducing the local pressure. The cycle repeats thanks to the

porous wick structure, where capillary action generates the liquid back-flow. Although HP can operate independently of gravity, when it is in a vertical orientation with the condenser facing upwards, gravity contributes to both the upward flow of vapour, due to the buoyancy force, and the backward flow of liquid, due to gravitational acceleration.

Conventional heat pipes have a cylindrical shape; however, to enhance the contact area with the batteries or cooling devices at the condenser section, different shapes are being investigated. There are two main alternative versions: the Flat Plate Heat Pipe (FPHP) and the Micro Heat Pipe Array (MHPA) [81]. FPHPs are obtained by flattening the classical cylindrical shape. This causes the collapse of the porous structure in the wick, increasing the hydraulic resistance encountered by the liquid flow. To overcome this limitation, the Vapour Chamber HP is emerging. This is a flat HP where the wick is present on both top and bottom surfaces, featuring wick columns that allow the condensed liquid to travel between them. A MHPA consists of a stack of several parallel micro HPs, forming a plate which increases the contact surface for flat shapes. Another emerging solution is the Pulsating Heat Pipe (PHP) [82], which is a wickless solution. In a PHP, the capillary effect is driven by the capillary dimensions of the channel, and, as sintered HP, they can work also without gravity [83]. Structurally, the PHP is a closed serpentine of a capillary tube, with the evaporator section on one side and the condenser section on the opposite side. Due to strong capillarity, surface tension effects are dominant, causing alternating liquid slugs and vapour plugs [81]. When the fluid expands at the evaporator section, it pushes the fluid towards the condenser. There, the vapour bubble shrinks, creating a pulsating, spring-like motion that continuously maintains fluid movement between the evaporator and condenser sections.

Without additional components, the only way to cool the condenser section is via natural convection. Greco *et al.* [84] developed a 1D thermal network model of a prismatic cell cooled by a HP and compared it with CFD simulations and an analytical approach. The three different approaches yielded reasonably similar results, validating the thermal network method. They subsequently employed the model to compare a HP using natural convection with the forced air cooling of a prismatic battery. The results showed a decrease of 23.9 °C in the maximum temperature when using the HP, which does not require power consumption unlike forced convection. Deng *et al.* [85] conducted experimental tests on a prismatic battery module composed of two cells in contact with a FHP featuring aluminium fins at the condenser section. They measured the temperature increase of the module during pulse cycle testing, observing a temperature roughly 5 °C lower compared to the HP without fins. The authors also pointed that the performance of heat pipes relying solely on natural convection depends significantly on the ambient temperature. Cattani *et al.* [86] conducted experimental tests under different discharge rates on a battery module composed of 9 cylindrical cells arranged in 3 rows and 3 columns, cooled by a PHP. In all cases, the two cells not in direct contact with the PHP (the one in the centre and another one in the side) exhibited the

highest temperatures. Liang *et al.* [87] developed a numerical CFD model coupled with a lumped electrochemical model to investigate the performance of a FHP for a full battery pack composed of 168 cells. This is one of the few works addressing heat pipe cooling on a full-scale pack. They found that the FHP alone is insufficient to ensure proper operating conditions for the battery pack.

Although HPs alone cannot manage an entire battery pack, they are a promising solution for integration with other methods, as they are passive, lightweight systems with high heat transfer coefficients.

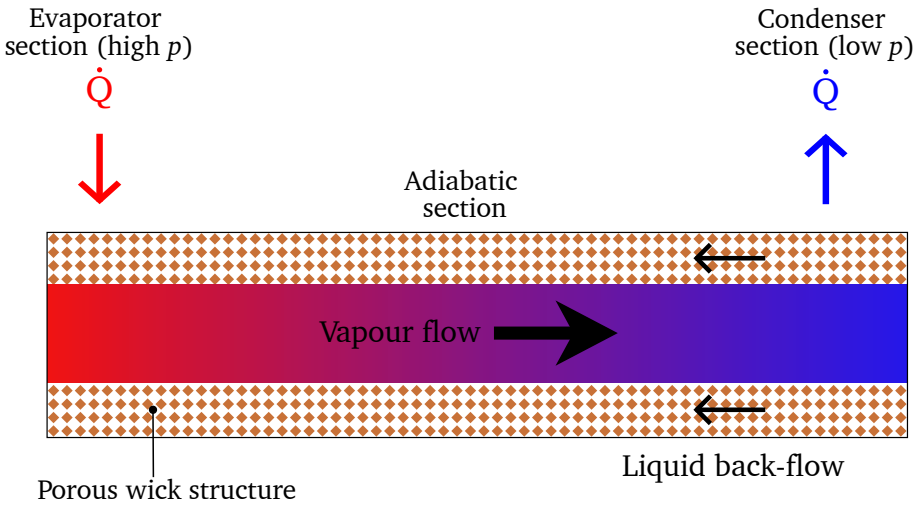


Figure 1.17: Sintered heat pipe working principle.

1.2.2.5. Hybrid BTMS

In their recent review, Bamdezh and Molaeimanesh [88] classified BTMSs into three main categories: active BTMSs, passive BTMSs, and hybrid BTMSs. Active BTMSs are the most commercialised, as they can ensure proper operating conditions; however, they consume power and offer inferior performance regarding temperature uniformity compared to passive BTMSs. The latter, beyond ensuring good temperature distribution, also possess the advantage of not requiring supplementary power. However, their limited heat transfer coefficients and significant dependence on ambient conditions make them unreliable for solely managing battery pack temperatures. Therefore, various studies are focusing on hybrid methods comprising an active system in conjunction with a passive system, aiming to reduce the power consumption associated with active systems while achieving the temperature uniformity provided by passive systems. The drawback is that hybrid systems are more expensive and complex compared to stand-alone active or passive systems.

Ranjbaran *et al.* [89] numerically investigated an air cooling system in combination with a PCM using different cooling duct structures and air stream pressure differences. They found that the longer the air ducts within the PCM, the more the battery surface temperature

decreased. Qin *et al.* [90] developed a numerical model, validated experimentally, of a hybrid BTMS featuring forced air cooling and PCM. They compared the proposed hybrid BTMS with PCM cooling alone and natural convection. They found that with forced air cooling at a high discharge rate (2C), the battery pack maximum temperature decreased by 35.2 °C and the maximum temperature difference within the pack decreased by 7.5 °C. Lv *et al.* [91] addressed the issue of bulky PCMs, which add weight and reduce the energy density of the battery pack. Therefore, they designed a hybrid BTMS where the PCM is composed of serpentine plates around the cells in the pack, combined with forced air cooling. This offers a lightweight structure and more space for airflow. They conducted a CFD simulation comparing the proposed hybrid system with a hybrid system featuring a classic PCM block. The results indicated that the new shape for the PCM reduced the maximum temperature and internal temperature difference, also leading to a $\sim$ 70% increase in the energy density of the pack.

An *et al.* [92] proposed a BTMS with a PCM integrated with expanded graphite and liquid cooling. In their system, cylindrical batteries are embedded within the PCM, in which vertical liquid channels are extracted; therefore, the liquid channels are not in direct contact with the cells. Thus, the heat generated within the cells is transferred through the PCM by conduction and then from the PCM to the coolant by convection. From CFD simulations, they concluded that the proposed hybrid BTMS can ensure good temperature uniformity even at a high discharge rate (3C). Xiao *et al.* [93] numerically analysed a BTMS with microchannel liquid cooling plates sandwiched between prismatic batteries and PCM. They highlighted the importance of parallel heat dissipation between the PCM and the coolant (*i.e.* both the PCM and the liquid plate must be in direct contact with the batteries) to meet cooling and heating requirements. They found that the presence of the PCM reduced the battery temperature by $\sim$ 4°C. They also conducted a failure analysis, which showed a 48% increase in the time required to reach the critical temperature in the event of liquid cooling failure; meanwhile, in the case of PCM failure, the critical temperature was not reached. Liu *et al.* [94] proposed a hybrid BTMS with PCM and copper foam in conjunction with helical pipes for liquid cooling to prevent the complete melting of the PCM. They found that by using helical channels embedded in the PCM, the heat transfer coefficient between the coolant and the PCM increased compared to conventional liquid channels.

Hybrid systems involving PCM are the most investigated in the literature; nonetheless, studies regarding hybrid systems with HPs have also emerged in recent years, where active cooling is used to cool the condenser section of the pipe. Tran *et al.* performed experimental tests to assess the performance of adding a FHP to a simple heat sink and to a system which combines a heat sink and forced air cooling. They found that by adding the HP, the thermal resistance of the heat sink was reduced by 30% under natural convection, and by 20% under forced convection. E *et al.* [95] validated a CFD model with experimental tests of a cylindrical

cell module cooled by HPs. These pipes were finned at the condenser section and cooled via forced air cooling. The proposed BTMS can maintain the maximum temperature and maximum temperature difference within the optimal range even at a 4C discharge rate. In addition, they found that the optimal fin spacing is 5 mm and the optimal fin thickness is 2 mm. Zhang *et al.* [96] proposed a different approach for HP and air cooling, where each HP has its own fan to cool the condenser section. Through experimental tests, they found that the proposed system can manage the battery temperature up to a 2C discharge rate, but it is insufficient from 2C onwards. For this reason, they proposed the addition of a thermoelectric cooler.

For higher performances, it is necessary to cool the heat pipe condenser with a higher heat transfer coefficient compared to the typical values offered by air cooling. Smith *et al.* [97] proposed a BTMS with two stages of heat pipes. The first stage removes heat from the battery, releasing it to an aluminium plate. The other side of the aluminium plate is in contact with the evaporator section of secondary heat pipes, which remove heat from the first stage and release it to an external liquid cooling plate. The advantages of this system are that it can improve temperature uniformity by $\sim$ 3 times compared to using only a cooling plate, and it avoids the presence of coolant inside the module. Furthermore, with the proposed system, the cooling plate can be downsized, requiring lower pumping power. Yuan *et al.* [98] experimentally analysed the cooling of a prismatic cell with an "L-shaped" HP, where one side of the "L" is in contact with the battery surface and the other side is in contact with a cooling plate placed below the cell. Compared to liquid cooling alone, the hybrid system lowered the maximum temperature by 2.1°C. Cattani *et al.* [99] proposed a hybrid BTMS combining two-phase direct liquid cooling with a PHP used to promote vapour recondensation. Through experimental tests, they found that the presence of the heat pipe could reduce the battery temperature by $\sim$ 15°C at a 2C discharge rate.

1.2.3. Air quality in BEV cabin

EVs, unlike ICEVs, do not produce exhaust emissions; for this reason, they offer superior cabin air quality. However, the thermohygrometric conditions of the cabin are fully managed by the HVAC system (including in heating mode, unlike ICE vehicles which utilise waste heat); consequently, the energy consumption of the HVAC system is higher in EVs, decreasing the vehicle range. The recirculation mode is commonly activated to reduce HVAC energy consumption. Rather than conditioning fresh air, this mode recirculates cabin air, which is closer to the set-point conditions since it has already been treated by the HVAC system. However, using recirculation decreases the dilution of pollutants inside the cabin. Therefore, ensuring good air quality in an EV cabin is an optimisation problem involving the trade-off between air quality and energy consumption. Vehicle cabins are confined spaces where the concentration of pollutants, CO_2, and pathogens can accumulate very rapidly.

According to Lesage *et al.* [100], using 100% recirculation mode during summer can extend the vehicle range by up to 40 km. They also proposed a control strategy for the blower flow rate depending on ambient conditions and other parameters such as the number of occupants and vehicle speed. Glos *et al.* [101] developed a Non-linear Model Predictive Control to assist the HVAC control system. For the specific case investigated, they found that fixing the cabin air quality set-point to 900 ppm (a value still within the acceptable range) can decrease energy consumption by 41.3% compared to the full fresh air case. Russi *et al.* [102] performed experimental measurements to characterise the air quality and temperature profile within an EV cabin. They compared the energy consumption of the HVAC system with recirculation off and on. During the initial $\sim$ 15 minutes, the cumulative energy consumption is almost identical for both modes. However, as time progressed, the effect of the recirculation mode became more evident; after one hour, the final energy consumption with recirculation mode is 70% of the consumption with recirculation off during heating, and 80% during cooling. Their work also demonstrated that the cabin air quality index increases significantly when the recirculation mode is disabled. Zong *et al.* [103] developed a multiscale model for air quality control in vehicle cabins. Their model relies on a 1D framework to evaluate the thermodynamic behaviour of the vehicle HVAC system and a 3D CFD model to obtain the cabin temperature distribution. Furthermore, they proposed a control strategy based on the non-uniform distribution of temperature inside the cabin and compared this with the classical control systems where a uniform cabin temperature is assumed. The proposed control strategy can save 9.1% of energy during summer and 18.8% during winter.

The literature indicates that predictive models are required to develop proper control strategies to ensure thermohygrometric comfort and air quality without excessive energy consumption. Most studies on this topic focus on CO_2 and pollutant concentrations; however, since vehicle cabins are small environments where passengers are in close proximity, the risk of infection due to respiratory droplets is high. Liu *et al.* [104] proposed a ventilation strategy for a car cabin to reduce the infection risk due to COVID-19, while ensuring thermohygrometric comfort and saving energy. The strategy relies on a dynamic model of the AC system, an infection risk assessment model, a predictive model for ambient conditions, and a ventilation control which considers passenger sneezing (the worst respiratory event in the context of droplet release). To save energy, fresh air blowing is activated only when necessary; therefore, a passenger voice recognition system for continuous sneezing is integrated into the control system to determine when to activate fresh air to reduce infection risk. Compared with regular ventilation, the strategy that integrates forced and regular ventilation reduces the infection risk by 19.2%. However, it increases energy consumption by only 0.15 kWh and the maximum cabin temperature fluctuation by 0.84 °C. When compared to the full ventilation strategy, the infection risk increases by 14.9%, but the energy consumption is 366.7% lower.

Finally, compared with the case without ventilation, the infection risk decreases by 65.9% with an increase of only 0.81 kWh in energy consumption.

Zhang *et al.* [105] proposed a novel impinging jet ventilation system, where the vents are strategically positioned in the cabin to ensure thermal comfort and reduce the risk of infection due to droplets released by coughing. They developed a numerical model to assess the performance of the proposed system. The results showed the effectiveness of the proposed strategy, reducing the droplet concentration near the breathing zone to almost zero in 20 s and promoting a uniform temperature distribution. Mariita *et al.* [106] proposed the integration of an Ultraviolet Subtype C (UVC) LED with the car cabin HVAC system to reduce the infection risk due to pathogens. This approach decouples thermal comfort management, performed solely by the HVAC system, from infection risk management, performed by the UVC LED. Therefore, unlike other strategies which push fresh air into the cabin changing the thermohygrometric conditions (which must then be re-established by the HVAC with extra power consumption), in this method, infection risk management does not influence the HVAC system energy consumption. From experimental measurements, they found that the HVAC system supplemented with UVC reduced pathogen levels significantly more than the HVAC alone, reducing the viral concentration in the cabin by over 90% in under 5 minutes.

1.3. Aim and scope

One of the primary limitations of BEVs is the thermal management of the battery pack, which restricts vehicle range and battery durability. High-energy batteries generate a large amount of heat within a limited volume, requiring a thermal management system capable of dissipating this heat while ensuring acceptable temperature uniformity. Consequently, the performance of the thermal management system defines the upper limit of the installable energy density and total battery capacity. In addition, the thermal management of BEVs is intrinsically linked to cabin air quality. Ideally, achieving optimal air quality requires a high air change rate; however, this imposes a significant energy drain on the HVAC system to bring fresh air to the set-point thermohygrometric conditions. To save energy, it is possible to use recirculation mode, but this negatively influences air quality. These aspects highlight the necessity of approaching the thermal design of BEVs from the earliest stages of vehicle conceptualisation. To achieve this, reliable and efficient numerical models are essential, as they permit the comparison of different thermal management strategies under various conditions at a contained cost relative to experimental testing. This does not imply that experimental tests are unnecessary; however, due to their high cost and limited flexibility, such tests are usually performed only when the design is largely fixed and are minimised to save resources. As highlighted in the previous section, the thermal design of a BEV involves different scales, ranging from the overall TMS (which also includes the thermal management of the motor, power electronics, and vehicle cabin) to the electrochemical

reactions within the battery cells that cause heat generation. As such, the use of high-fidelity 3D models, such as CFD or FEA, to model the entire system is not a feasible approach, as it would require an excessive amount of resources and time. Therefore, 1D and lumped parameter models represent the best trade-off between computational cost and accuracy for modelling overall systems. Thanks to the flexibility of these reduced-order models, it is possible to easily combine local phenomena with a global description of the domain. Models that combine phenomena at different scales are known as "multiscale models". To develop multiscale models, it is fundamental to use numerical tools capable of interfacing the sub-models used for each distinct scale. To increase the accuracy of these reduced-order models, specific parameters of sub-components can be extracted from sub-models based on high-order methods (by modelling only the sub-component) or from experimental data when available. This thesis presents multiscale methods and a reduced-order optimisation approach for the thermal design of BEVs.

The next chapter presents the development of a numerical tool for the reduced-order modelling of pipe flow networks and thermal networks. The numerical tool is based on a physical domain description using graph theory, which discretises the domain as a network of nodes and branches. Here, the nodes are used to lump the mass of the referred components, and the branches are used to model the general fluxes between the nodes. The solvers developed within this framework are based on balance equations, where the source terms are formulated using a linear polynomial, and potential non-linearities are treated by employing iterative procedures. When solving the equations, the solvers check for dimensional consistency to avoid erroneous results due to the incorrect use of units for the input parameters.

The thermal network solver is subsequently employed to develop a multiscale model to test the thermal behaviour of an integrated system composed of two battery packs and a DC-DC power converter for a BEV. The integration is achieved by means of a cooling plate which serves as structural basis and cools both the batteries and power converter components. This system is tested under a real driving scenario based on the Worldwide Harmonised Light Vehicle Test Procedure (WLTP). The multiscale model begins at the electrochemical reaction scale, described with an electrochemical model, to determine the total heat generation within the cells. The cell description is then scaled to the module level; due to the high number of individual cells in the overall system, each node in the thermal network describes a module rather than a single cell, to reduce the computational cost. The heat transfer between the modules and the coolant is described by employing the convective resistance of the cooling plate. This resistance is extracted using a scaled CFD model, where only a periodic portion of the plate is simulated under the fully developed flow assumption, rather than modelling the entire volume. Therefore, the developed multiscale model interfaces the overall system, composed of two packs and a power converter, with the heat transfer at the module level

and heat generation within individual cells and power switches. This analysis is part of the European project SCAPE, which aims to set new standards for efficiency and high power density in power electronic components for EVs.

The liquid cooling plate proposed to cool the packs and power converter includes two parallel serpentines. This configuration is standard and has been employed only to carry out a preliminary evaluation of system performance. To improve the cooling capability and temperature uniformity, a reduced-order optimisation method is developed. The optimisation is performed on the path of the channels inside the cooling plate to improve the standard serpentine path. The optimisation tool is based on a network-like description of the channel, where an empty grid of nodes serves as the starting point from which new paths are constructed. The paths are generated using Ant Colony Optimisation (ACO), a probabilistic meta-heuristic algorithm inspired by the natural behaviour of ant colonies searching for the best path to food sources. To the best of the author's knowledge, this is the first application of ACO to optimise the topology of cooling channels. The evaluation method for new solutions is based on the thermal network solver relying on empirical correlations for the convective heat transfer coefficient and friction factor.

In the final chapter of the thesis, the problem of air quality is addressed via a multiscale model to calculate the trajectories of exhaled respiratory droplets under five different scenarios: mouth breathing, nose breathing, speaking, coughing, and sneezing. The multiscale model is based on coupling an analytical model to describe the trajectory of droplets (with diameters ranging from 1 mm to 0.1 μm) with a 3D CFD model used to describe the airflow features at the macro-scale induced by the specific respiratory event and the local ambient conditions required by the analytical model to calculate droplet trajectories. Therefore, this multiscale model exploits the flexibility and accuracy of an analytical tool alongside the capability of 3D CFD simulations to describe complex airflow characteristics which are not addressable with a meshless analytical approach. From the results on droplet trajectories, and including an inhalation model, the infection risk due to airborne inhalation and fomite transmission is assessed through a statistical analysis based on thousands of runs for each scenario. To validate this novel modelling approach before applying it to complex geometries, the methodology is demonstrated on a basic test case involving an infected individual in a general indoor environment. Although the test case is not fully representative of a vehicle cabin scenario, it provides insights into the mechanisms of pathogen transmission and permits a comparison of different respiratory events.

1.4. Organisation of the thesis

The thesis is divided into four chapters, excluding the present one and the chapter containing the concluding remarks. Each chapter starts with a brief introduction and a literature survey of the topic discussed.

Chapter 2 presents the development of the numerical tool for the solution of pipe and thermal networks. This chapter discusses in detail the mathematical domain description based on graph theory, the modelling of source terms and common circuit components, the management of units of measurement, and the solver implementation including the derivation of balance equations.

Chapter 3 presents a multiscale model of an integrated system comprising battery packs and a power converter cooled by a cooling plate for a passenger BEV. Following the presentation of the case study, the sub-models interfaced by the multiscale model are discussed. Subsequently, the model is applied to assess the operating temperature of the components under a WLTP drive cycle.

Chapter 4 details the reduced-order meta-heuristic optimisation of cooling channel paths. The chapter describes the implementation of the ACO algorithm, outlining the principal variations that have emerged over the years. The algorithm selected for this specific application is then discussed and applied to optimise the cooling plate used in the preceding chapter to cool the integrated system.

Chapter 5 addresses air quality and the infection risk associated with respiratory droplets. The multiscale model used to perform the statistical analysis is presented and applied to assess the infection risk of different respiratory events in an indoor environment. The results of each event are compared, highlighting in detail the differences and implications.

Finally, after the conclusions, Appendix A derives and explains the numerical methods used to develop the numerical tool and to perform the time integration of the voltage equation used in the electrochemical model employed to calculate heat generation within the battery cells.

Chapter 2

Development of a numerical tool for reduced-order analysis of thermo-fluid systems

NOMENCLATURE I

Latin letters

$\boldsymbol{A}$	Incidence matrix
$\boldsymbol{B}$	Upwind matrix operator
C	Heat capacity [$\mathrm{J\,K^{-1}}$]
C_r	Heat capacity rate [$\mathrm{W\,K^{-1}}$]
$\boldsymbol{C}$	Diagonal matrix of the heat capacity rates of branches [$\mathrm{W\,K^{-1}}$]
c_p	Specific heat [$\mathrm{J\,kg^{-1}\,K^{-1}}$]
$\boldsymbol{c}$	Vector of coefficients associated with the SI base units
D_h	Hydraulic diameter [m]
$\vec{d}$	Euclidean distance vector [m]
$\boldsymbol{d}$	Vector of unsteady terms in the continuity equation
$\dot{E}_v$	Mechanical losses due to viscous effects [W]
e	error
$\boldsymbol{e}$	Vector of exponents associated with the SI base units
f	Equivalent Darcy friction factor
G	Thermal conductance [$\mathrm{W\,K^{-1}}$]
g	Gravitational acceleration [$\mathrm{m\,s^{-2}}$]
H	Volumetric heat generation coefficient [$\mathrm{W\,m^{-3}}$]
h	Heat transfer coefficient [$\mathrm{W\,m^{-2}\,K^{-1}}$]
$\boldsymbol{I}_\mathrm{bou}$	Mask of boundary nodes
$\boldsymbol{I}_\mathrm{int}$	Mask of internal nodes
K	Concentrated loss coefficient
k_σ	Constant source coefficient
$\boldsymbol{k}$	Vector of know terms in the mechanical energy balance [Pa]
L	Length [m]
$\boldsymbol{L}$	Matrix of coefficients in the thermal network solver [$\mathrm{W\,K^{-1}}$]
$\boldsymbol{M}$	Diagonal matrix of elements $\rho_j S_j$ [$\mathrm{kg\,m^{-1}}$]
m_σ	Linear source coefficient
$\boldsymbol{N}$	Matrix operator of sum over branches
n	Amount of
$\vec{n}$	Normal unit vector
$\boldsymbol{O}$	Orientations of branch velocities
P	Total fluid pressure [Pa]
p	Fluid pressure [Pa]
$\dot{Q}$	Heat [W]
q	Heat flux [$\mathrm{W\,m^{-2}}$]
R	Thermal resistance [$\mathrm{K\,W^{-1}}$]
$\boldsymbol{r}$	Vector of known terms in the thermal network solver [W]
Re	Reynolds number
S	Surface area [$\mathrm{m^2}$]
$\boldsymbol{s}$	Vector of known terms in the thermal energy balance [W]
T	Temperature [K]
t	Time [s]
U	Overall heat transfer coefficient [$\mathrm{W\,m^{-2}\,K^{-1}}$]
u	Streamwise velocity component [$\mathrm{m\,s^{-1}}$]
V	Volume [$\mathrm{m^3}$]
$\vec{v}$	Fluid velocity [$\mathrm{m\,s^{-1}}$]
$\boldsymbol{W}$	Matrix of coefficients in the thermal energy balance [$\mathrm{W\,K^{-1}}$]
W	Mechanical power provided by pumps and fans [W]
$\vec{x}$	Position in a Cartesian system
$\boldsymbol{Y}$	Matrix of velocity coefficients in the mechanical energy equation [$\mathrm{Pa\,s\,m^{-1}}$]
z	Gravimetric height [m]

NOMENCLATURE II

Greek letters

γ	Conversion coefficient
ε	Effectiveness
ε_r	Emissivity
λ	Thermal conductivity $[\mathrm{W\,m^{-1}\,K^{-1}}]$
ν	Kinematic viscosity $[\mathrm{m^2\,s^{-1}}]$
ρ	Density $[\mathrm{kg\,m^{-3}}]$
σ	Generalised source term
σ_B	Stefan Boltzmann constant $[\mathrm{W\,m^{-2}\,K^{-4}}]$
τ	viscous stress tensor
ϕ	Flow rate coefficient
φ	Generic field
χ	Generic numerical value
ψ	Head coefficient

Subscripts

b	Branches
E	Egress
ext	External
I	Inlet
i	Node i
int	Internal
j	Branch j
L	Losses
n	Nodes
P	Pump
ud	Unidirectional

Superscripts

n	n-th time-step

2.1. Introduction

BEVs integrate multiple subsystems whose interactions strongly influence vehicle performance, safety and durability. Assessing these interactions already in the design phase is essential, not only to evaluate the behaviour of individual components, but also to understand how their coupling affects the operation of the complete system. High-order numerical simulations provide detailed insight but are rarely suitable for full-system analyses because of their computational demands. Consequently, system-level studies typically rely on lumped-parameter models, which allow the designer to explore alternative architectures, operating conditions and thermal strategies within reasonable time. Commercial software packages are available for building lumped-parameter models, and many of them offer user-friendly interfaces that facilitate rapid model construction. However, their internal formulations and numerical procedures are generally not accessible to the user, which restricts the possibility of modifying or extending the underlying models. As a result, their applicability is often confined to predefined component libraries or specific classes of problems, limiting their use when non-standard system architectures, customised physical models or tailored numerical strategies are required.

In this context, the work presented in this chapter introduces a computational framework designed for the formulation and analysis of lumped-parameter models of thermo-fluid systems. The methodology has been conceived with a broad level of generality, aiming at providing a flexible modelling environment that can be adapted to a wide range of engineering problems. This general structure is also intended to enable future integration

with higher-fidelity numerical tools when more detailed investigations of specific components or phenomena are required.

2.2. Literature survey

Simulink [107] is among the most widely adopted platforms for modelling and simulating complex systems, largely due to its intuitive block diagram interface based on the signal port approach, which facilitates the construction of multi domain models involving different physical phenomena. Thermo-fluid problems are addressed through the Simscape Fluids libraries, which provide sets of differential and algebraic equations describing one dimensional flow. Fluid properties and energy transport are evaluated using upwind and smoothed upwind numerical schemes, while streamwise heat conduction is represented through an equivalent thermal conductance. The software also incorporates sparse matrix solvers, which are automatically selected when appropriate for the system under analysis. Time integration can be carried out with either the backward Euler method or the trapezoidal rule, although backward Euler is always enforced at the beginning of a simulation and whenever discrete internal events occur within a component. Simulink also includes a variable step solver, which adjusts the time-step according to the model states: when rapid changes occur, the solver decreases the step size to increase accuracy, while during slow variations the step size is increased to avoid unnecessary computations. Like Simulink, there are other softwares that provide a diagram block interface, such as AMESim [108] and GT-SUITE [109]. AMESim is a commercial software commonly used for the simulation of multi domain physical systems, especially in the automotive and aerospace sectors. The software is written mainly in C and C++ and relies on a set of predefined component libraries whose behaviour is described by systems of differential and algebraic equations representing lumped physical models. Users build models by connecting these components through standardised physical ports, while the internal implementation of the equations remains hidden. The thermo-fluid libraries provide models for single and multi-phase one dimensional flow, heat transfer and phase change, and rely on empirical and semi-empirical correlations. Time integration is handled by a selection of explicit and implicit solvers, automatically chosen by the software depending on stiffness, discontinuities and the presence of algebraic constraints. Event handling algorithms adapt the time-step and reinitialise internal states when abrupt changes occur. GT-SUITE [109] is a general purpose simulation platform, implemented mainly in C++, originally developed for engine applications and now widely used for full vehicle modelling, including lubrication circuits, thermal systems and battery cooling. Models are built from libraries of components that represent control volumes, flow restrictions, heat exchangers and mechanical elements. These components are based on one dimensional conservation equations complemented by empirical correlations for transport and heat transfer processes. The numerical solver uses implicit integration schemes that can handle strongly coupled thermo-fluid systems with stiff

dynamics. Adaptive time-stepping and automatic event detection are employed to maintain stability and accuracy in the presence of discontinuities such as valve operations or phase transitions. The software also includes sparse matrix solvers and supports both transient and steady-state simulations. One of the main strengths of GT-SUITE is the integration of high fidelity three dimensional component models into one dimensional and zero dimensional system level models. This allows the use of detailed multi physics boundary conditions and enables two way interaction between subsystems. GT-SUITE is widely recognised in the industry for powertrain and engine simulations and includes tools for applications ranging from combustion to NVH (noise, vibration and harshness) analysis. Like GT-SUITE, other software integrates the CAD model directly into the user interface to support the conceptualisation of system models. An example is Altair Flow Simulator [110], a tool for modelling systems based on fluid flow and heat transfer represented as networks. It can solve steady-state and transient flows, both compressible and incompressible, through networks of pipes, ducts, chambers and components, fully coupled with a thermal network. The software includes an extensive library of predefined elements with validated pressure drop and heat transfer data, and it is also able to model combustion processes. Its solver assembles the mass, momentum and energy equations of the network and solves them using the Newton-Raphson algorithm. Although Altair Flow Simulator is a commercial tool, it offers a high level of expandability thanks to the user defined elements, which allow users to implement custom formulations of the mass, momentum and energy equations through Python [111] or Fortran interfaces. In the field of renewable energies, TRNSYS [112] is one of the most widely used tools for simulating transient systems. Its applications include energy systems such as solar thermal installations, photovoltaic systems, wind turbines and HVAC. As in Simulink, the model is constructed by connecting different components, where each component is a mathematical function (written in FORTRAN) with its own inputs and outputs, and the links define the flow of data. TRNSYS can solve differential equations either numerically, for example with second order Runge-Kutta methods, or analytically when this is possible. The default strategy for solving the full system is the successive substitution method, while for systems that are strongly non linear or when backsolving issues occur, Powell's method is preferred. In the field of aerospace, one of the first software tools used as a standard is SINDA/FLUINT [113]. It was originally developed in FORTRAN as a general thermal analyser for spacecraft, capable of solving only thermal networks. Over time, it evolved into a general lumped parameter network solver for thermo-fluid problems, handling steady and transient flows with both incompressible and compressible behaviour, and showing strong performance for multiphase phenomena. The thermal and fluid domains are treated separately and are linked through convective resistances. Time integration can be performed with either explicit schemes, such as forward difference, or implicit schemes, such as Crank Nicolson, with an option for automatic time-step adjustment. Currently, SINDA/FLUINT is no longer a stand alone

software but is integrated into other tools that provide a graphical user interface, such as Thermal Desktop [114].

All the software cited above is distributed under closed licenses. They are typically characterised by well established workflows, intuitive graphical interfaces, and highly integrated modelling environments. However, their internal formulations and solver structures are not accessible to the user, which limits the ability to verify numerical assumptions or adapt the tools to unconventional modelling problems. In parallel, an open source software makes its implementation fully available, which enables reproducibility and allows customisation for problems that differ from the original purpose of the tool. For these reasons, this kind of software is often preferred in research contexts. The main limitation is the higher level of user expertise required, due to the lack of user friendly interfaces and complete documentation. Among the open source tools, OpenModelica [115] is one of the most widely used modelling and simulation environments for both academic and industrial applications. The default solver is based on DASSL [116, 117], which discretises the differential equations with a backward differentiation formula and then applies Newton method to solve the non linear system iteratively. To speed up convergence, the initial guess for Newton method is obtained from a polynomial that interpolates the function values from previous time-steps. This solver is the standard choice thanks to its implicit structure, step size control, and high order. Since OpenModelica is open source, the code includes many solvers added over the years by developers. The most used ones, besides DASSL, are based on Euler, Newton-Raphson, and Runge-Kutta numerical methods. OpenModelica belongs to a family of modelling and simulation software (such as Dymola [118]) written in the Modelica language [119], an object oriented declarative language developed by the Modelica Association, a non profit organisation. Its main characteristic is its a-causal formulation, in which the model is specified through equations rather than explicit variable assignments. The Fluids library provides components for one dimensional thermo-fluid flow. It is designed to be general enough to model both incompressible and compressible flows, with one or more phases. Another well known open source software is Scilab [120], originally developed at INRIA and now maintained by the Scilab Consortium. Unlike Modelica, the Scilab language is closer to classical numerical languages such as MATLAB. Scilab is written mainly in C and C++, and it provides a broad set of numerical routines for linear algebra, optimisation, ODE and DAE integration, and signal processing. Most of the available integrators are implicit, such as Lsodar [121], CVODE [122], and IDA [123], while the explicit solvers consist mainly of Runge-Kutta and Dormand-Prince schemes. To support system modelling, the Scilab distribution also includes Xcos, a graphical editor that allows users to build virtual systems by connecting blocks that represent functions.

Most of the thermo-fluid modelling tools presented were originally developed in FORTRAN or, in more recent generations, in C or C++. This prevalence is partially explained by the

historical dominance of these languages in scientific computing. More importantly, both FORTRAN and C++ still offer highly optimised numerical performance, predictable memory management, and direct access to mature linear algebra libraries, which are essential for solving large coupled systems of equations. Nowadays Python [111] has become one of the most widely adopted languages in modern research environments, largely because of its intuitive and simple syntax, comparable to pseudocode, and its shorter development cycle compared to compiled languages such as C++ and FORTRAN. Python offers dynamic typing, automatic memory management, and no requirement for explicit variable declarations, which reduces the amount of low level programming needed to implement numerical algorithms. The ability to convert or reinterpret data structures without rigid type constraints also simplifies prototype development and methodological experimentation. As a result, new model features, alternative discretisation approaches, or entirely new solver strategies can be implemented and tested more rapidly. However, the same features that make Python easy to develop lead to longer execution times compared to compiled languages. In pure Python, numerical loops and iterative solvers usually run slower because the interpreter processes each operation at runtime. This limitation is largely mitigated in scientific programming through the use of optimised libraries such as NumPy [124] and SciPy [125], whose core numerical routines are implemented in C, C++, and FORTRAN. By expressing computations in vectorised form or by delegating linear algebra, sparse matrix operations, and integration routines to these compiled libraries, Python programs can achieve performance levels close to traditional compiled codes while retaining the advantages of a high level and flexible development environment. For these reasons, Python provides an effective balance between development efficiency and computational performance, making it well suited for modern scientific modelling workflows. An example of modelling software developed in Python is PyPSA (Python for Power System Analysis) [126], which is used for modelling and simulating modern power energy systems by creating networks of sources, loads, and distribution components. The solution is obtained by solving the energy balance equation through the Newton-Raphson method. Another well known Python package is TESPy (Thermal Engineering Systems in Python) [127], which models thermal systems using a network based representation and assembles the mass, momentum, energy, and composition balance equations for all components in the network, solving them also with the Newton-Raphson method. By default, fluid properties in TESPy are obtained through the CoolProp library [128], an open source thermophysical property library that provides accurate and high fidelity property calculations for a wide range of pure fluids and mixtures. Although CoolProp is developed in C++, it includes wrappers for Python, MATLAB, Modelica, and other languages, which allows for easy integration with most of the software tools used in the scientific modelling community.

2.2.1. Methodology

In this section, the methodology applied to develop the lumped parameter computational tool, object of this chapter, is presented. All the solvers implemented in the library are based on three core pillars which define the strategy applied to discretise the domain, assembly the balance equations and manage the units of measurement.

i. The computational domain is represented by a graph consisting of nodes connected by directed and weighted branches. Each node is defined as a point in space with its own coordinates and a unique label, which may be an integer index or a text string. Distinct labels are essential because branches are defined by their start and end nodes; therefore, to distinguish one branch from another, all nodes must have unique identifiers. As nodes represent the mean value of the variables within the region they represent, they should be placed where zero or small gradients are expected. Parallelly, branches represent the paths through which fluxes are exchanged. During the construction of the graph, all the information on nodes and branches is stored in dedicated containers. Once the topology of the domain is fully defined, this information is used to assemble the incidence matrix. This matrix provides a mathematical description of the graph connectivity and forms the basis for building the system of equations for the investigated problem.

ii. The source and sink terms of the governing equations are always generalised as linear polynomials, where the variable is the value of a specific entity (node or branch) in the field associated with the balance equation under consideration. The known term of the polynomial is a constant value. In many cases, the source/sink term is non-linear and cannot be described by a linear equation. In such instances, the non-linear terms are incorporated into the known term using the value from the previous step. Iterative methods are then employed to determine the value for the current step.

iii. Since the solvers operate with dimensioned quantities, the tool includes a unit management system. Units are defined using two arrays: one stores the exponents of the SI base units to identify the type of each physical quantity, and the other stores the coefficients that scale the base units. This enables users to input parameters directly using standard prefixes from the reference data. All quantities are internally converted to SI base units through the coefficient array and returned in the units provided by the user. The unit system is not only applied to individual variables, but also to the arrays and matrices assembled by the solver. This enables unit consistency to be checked during computation by comparing the exponent arrays of all terms in the system of equations.

These core aspects were used to develop solvers for one-dimensional flow in pipe networks and thermal networks. The one-dimensional flow solver is based on a staggered Finite Volume

approach, in which the topology is divided into control volumes according to two schemes: one for the nodes and one for the branches. A node's control volume includes the volume associated with the node itself and half of the volume of each connected branch. The control volume of a branch corresponds to the fluid volume represented by that branch. The mass and energy balance equations are solved on the node control volumes, while the mechanical energy balance is solved on the branch control volumes. A variant of the SIMPLE algorithm [129] is used to decouple the pressure and velocity fields for both steady-state and transient problems. The effects of components commonly found in pipe networks, such as pumps and valves, are modelled as linearised source terms in the mechanical energy balance.

The thermal network solver is based on the electro-thermal analogy, in which heat transfer problems are represented by a network of thermal nodes and resistances. This type of discretisation fits naturally with the graph representation: thermal nodes correspond to graph nodes, representing the mean temperature of lumped regions, while thermal resistances correspond to branches representing paths through which heat transfers between nodes. The solver computes the values of the thermal resistances iteratively, enabling the method to account for their dependence on temperature, which may be caused by fluid properties, natural convection, or thermal radiation. No specific numerical scheme is required for the steady-state problem, as the solution can be obtained analytically. The only exception occurs when the resistances or source terms depend on temperature; in this case, the system is solved using iterative procedures.

The library is implemented in Python 3 [130], which, as discussed in the literature review, strikes a good balance between flexibility, syntax complexity, and computational performance.

2.3. Domain discretisation

The computational domain is represented by a directed and weighted graph composed of nodes connected by directed branches. Each node is a point in space with generic coordinates defined as:

$$\vec{x} = (x_1, x_2, x_3)\,. \tag{2.1}$$

In addition to its spatial coordinates, each node is assigned a unique identifier, or *key*. This key can be either an integer value or a text string, and it is primarily used for defining the boundary conditions and sources.

Each branch is identified by its starting node $\vec{x}_{\mathrm{s}}$ and its ending node $\vec{x}_{\mathrm{e}}$. Although the branches are uniquely defined by their connected nodes, a *key* is also assigned to each branch to facilitate the easy definition of sources or other specific properties on them. The direction and the length of each branch are calculated as follows:

$$\vec{d} = (x_{1,\mathrm{e}} - x_{1,\mathrm{s}},\, x_{2,\mathrm{e}} - x_{2,\mathrm{s}},\, x_{3,\mathrm{e}} - x_{3,\mathrm{s}})\,, \tag{2.2}$$

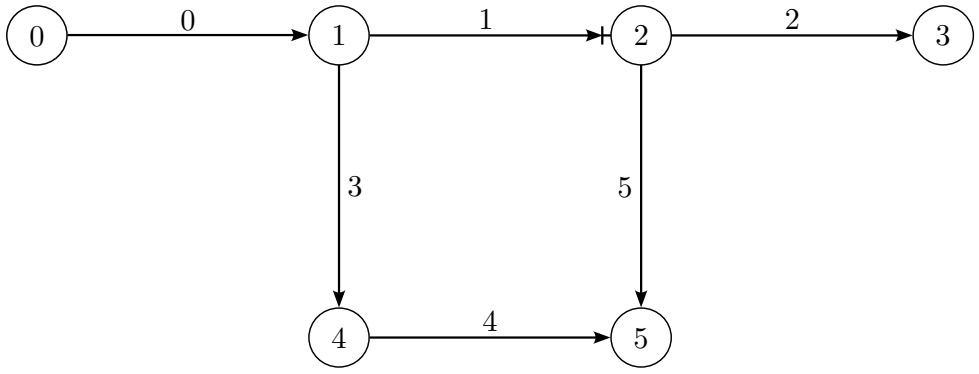


Figure 2.1: Example of a simple directed graph, where nodes and branches are identified by integer indices.

$$L = |\vec{d}| \,. \tag{2.3}$$

Solving the governing equations often requires information that cannot be directly derived from the graph structure. For example, in problems related to pipe flow, geometric features such as the hydraulic diameter and the cross-sectional area are necessary. To accommodate this, a dedicated container is assigned to each branch, allowing for the storage of any additional information required by the specific solver. Figure 2.1 illustrates a network example with six nodes and six branches, where the arrows indicate the direction of the branches. Mathematically, the connectivity of the graph is described by the incidence matrix $\boldsymbol{A}$. This matrix has a number of rows equal to the number of nodes in the graph and a number of columns equal to the branch count. Each element A_{ij} of the incidence matrix $\boldsymbol{A}$ is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if node } i \text{ is the starting point of branch } j \\ -1 & \text{if node } i \text{ is the ending point of branch } j \\ 0 & \text{if there is no interaction between node } i \text{ and branch } j \end{cases} , \tag{2.4}$$

or, alternatively:

$$A_{ij} = \begin{cases} 1 & \text{if branch } j \text{ originates from node } i \\ -1 & \text{if branch } j \text{ terminates at node } i \\ 0 & \text{if there is no interaction between branch } j \text{ and node } i \end{cases} . \tag{2.5}$$

By applying Equation (2.4) (or equivalently, Equation (2.5)), the incidence matrix represent-

ing the graph in Figure 2.1 is given by:

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \end{pmatrix}. \tag{2.6}$$

The graph defined by Equations (2.4) and (2.5) features bidirectional branches, meaning that the generic fluxes across these branches can flow in either direction: from the starting to the ending node, or vice-versa. This behaviour is correct for most physical systems addressed; however, there are cases where a branch must be unidirectional. An example is the modelling of a check valve, which permits the fluid flow only along one direction. When the graph includes a connection with a unidirectional branch (such as branch 1 in Figure 2.1), the following rule is applied:

$$A_{\mathrm{ud},ij} = \begin{cases} 1 & \text{if node } i \text{ is the starting point of unidirectional branch } j \\ 0 & \text{if node } i \text{ is the ending point of unidirectional branch } j \\ 0 & \text{if there is no interaction between node } i \text{ and branch } j \end{cases}. \tag{2.7}$$

To account for the presence of unidirectional branches, a variant of the incidence matrix, denoted $\boldsymbol{A}_{\mathrm{ud}}$, is defined by combining Equations (2.4) and (2.7).

$$\boldsymbol{A}_{\mathrm{ud}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \end{pmatrix} \tag{2.8}$$

In most cases, $\boldsymbol{A}$, and equivalently $\boldsymbol{A}_{\mathrm{ud}}$, is a *sparse matrix*, a matrix where the majority of its elements are zero. This characteristic arises because nodes are typically only connected to a small subset of other nodes, resulting in no interaction with non-directly connected nodes. This property allows $\boldsymbol{A}$ to be stored using the *Compressed Sparse Row (CSR)* scheme, which saves memory by not storing elements with a value of zero. Specifically, the CSR format consists of three arrays:

i. The values array, which stores all the non-zero elements of $\boldsymbol{A}$.

ii. The column indices array, which stores the column index for each corresponding value

in the first array.

iii. The row pointers array, which contains integers that map the start and end of each row's data within the first two arrays.

The mapping is achieved by creating a range of integer indices that runs from the value at the i-th element of the row pointers array, up to the value at the $(i+1)$-th element of the same array. The value at the $(i+1)$-th element is excluded from the range. This range of indices represents the positions in the values and column indices arrays that belong to the i-th row. Table 2.1 shows the three arrays for the matrix in Equation (2.8). For instance, to retrieve the elements corresponding to the second row in $\boldsymbol{A}_{\text{ud}}$ (which has index 1, following Python's zero-based enumeration), one creates a range from the value at index 1 of the row pointers array, up to the value at index 2 (excluded). The resulting index range is $[1, 4)$, which corresponds to array indices $1, 2$, and 3. This means that the elements in the values array at these indexes (*i.e.*, -1, 1, and 1, respectively) are the non-zero elements of the second row.

In addition to optimal memory management, the CSR format offers superior performance compared to the classic dense array format. This advantage stems from the use of dedicated, fast algorithms for operations such as inner products between matrices and vectors, and the solution of linear systems. The main drawback of the CSR format, compared to the classic array format, is the difficulty in modifying or adding elements due to its static definition. The *SCiPy* library [125], utilised in the numerical tool described in this chapter, provides the necessary data structures and algorithms for sparse linear algebra.

2.4. Dimensioned values and units of measurement

The solvers developed within the present toolbox utilise dimensional equations, meaning that every term in each governing equation must have the same physical dimension. To correctly handle these dimensional equations and prevent errors caused by unit inconsistency, a robust units of measurement management system is included in the library. The failure of NASA's Mars Climate Orbiter [131] serves as a notable example of how incorrect or inconsistent use of measurement units can lead to adverse consequences. In that case, the mission failed because one part of the ground software calculated the impulse in Pound-force seconds, while another part expected the value in Newton seconds. This discrepancy resulted

Table 2.1: Compressed Sparse Row (CSR) format of the matrix in Equation (2.8).

values	1	−1	1	1	−1	1	1	−1	−1	1	−1	−1
column indices	0	0	1	3	1	2	5	2	3	4	4	5
row pointers	0	1	4	7	8	10	12					

in incorrect trajectory calculations, leading to the destruction of the orbiter upon entering Mars' atmosphere.

The units of measurement manager in this toolbox is designed to execute operations between dimensioned values. Its primary scope is twofold: it checks the consistency of the units between the operands before performing an operation, and it correctly defines the propagation of the units from the operands to the result, thereby ensuring the output maintains the physically correct dimension.

2.4.1. Internal Unit Representation

The treatment of units of measurement is based on the use of two internal arrays: one array for the exponents that define the physical quantity ($\boldsymbol{e}$), and another array for the coefficients that define the multiplier of the base unit ($\boldsymbol{c}$). Each array includes seven elements, with each element associated with one of the seven units defined by the International System of Units (SI): `Kg` (mass), `m` (length), `s` (time), `K` (temperature), `mol` (molar mass), `A` (current intensity), and `cd` (luminous intensity). The exponent and coefficient arrays are defined as:

$$\boldsymbol{e} = \left(e_{\mathrm{kg}}, e_{\mathrm{m}}, e_{\mathrm{s}}, e_{\mathrm{K}}, e_{\mathrm{mol}}, e_{\mathrm{A}}, e_{\mathrm{cd}}\right) , \tag{2.9}$$

$$\boldsymbol{c} = \left(c_{\mathrm{kg}}, c_{\mathrm{m}}, c_{\mathrm{s}}, c_{\mathrm{K}}, c_{\mathrm{mol}}, c_{\mathrm{A}}, c_{\mathrm{cd}}\right) , \tag{2.10}$$

where each subscript refers to the corresponding SI base unit. For each non-zero element in the exponent array, the physical quantity receives a contribution from that element, scaled by its specific coefficient from the coefficient array. For instance, to define the unit `centimeter per second` ($\mathrm{cm\,s^{-1}}$), the exponent array is defined to show the dependency on length and time, while the coefficient array includes the necessary conversion factor. The arrays for $\mathrm{cm\,s^{-1}}$ are:

$$\boldsymbol{e} = (0, 1, -1, 0, 0, 0, 0) , \tag{2.11}$$

$$\boldsymbol{c} = (1, 0.01, 1, 1, 1, 1, 1) . \tag{2.12}$$

In the exponent array, the second and third elements are 1 and -1 (for length and time, respectively), while all other elements are zero. In the coefficient array, the second element is 0.01 (to convert meters to centimeters), and the third element is 1. By default, coefficients corresponding to elements where the exponent is zero are set to 1.

2.4.2. Unit Definition and Parsing

In the toolbox, a unit of measurement instance can be created either by directly inserting the exponent and coefficient arrays or by using a text string. The string format follows the pattern:

```
<c1><u1>^<e1> <c2><u2>^<e2> ...
```

where <c> is the coefficient, <u> is the base unit symbol, and <e> is the exponent. The coefficient, represented by <c>, can be inserted either as a text prefix, following the symbology defined in Table 2.2, or as any numerical value. Using a numerical value allows the direct inclusion of factors other than standard multiples of ten. The text string for cm s^{-1} is simply

```
''cm s^-1''        or        ''0.01m s^-1'' .
```

This text string is then translated into the exponent and coefficient arrays through a recursive parser.

Once the seven base units are defined, all other derived units can be represented. For instance, the SI derived units for energy (`J`) and pressure (`Pa`) can be written in terms of base units as:

```
J:      kg m^2 s^-2
Pa:     kg m^-1 s^-2 .
```

Since the problems analysed by this library involve many derived units, a pre-stored array is used to associate the definition of the most common derived units (such as Joule, Pascal, and Watt) with their base unit strings. In this way the Pascal can be defined simply as `Pa`. The parser will then search for its base unit definition in the storage and translate it into the corresponding exponent and coefficient arrays, effectively mapping `Pa` to `kg m^-1 s^-2 .`

2.4.3. Operations Between Dimensioned Values

Once the internal representation of units and the methods for their creation are established, the next crucial step is to define their behaviour during operations between dimensioned values. While the numerical calculations involving scalars, vectors, and matrices are handled by the native features of the programming language (Python, in this context), the unit propagation must be explicitly managed. This explicit unit management is necessary to prevent calculations involving incompatible units and to ensure the resultant physical dimension is correct.

Before any operation, all involved units are first converted to their SI representation. Once this conversion is complete, the determination of the output unit simplifies to providing the correct resultant exponent and coefficient arrays.

Table 2.2: Symbols and values associated with the multiples and sub-multiples of the SI system.

sub-multiples	symbol	`p`	`n`	`mu`	`m`	`c`	`d`
	value	10^{-12}	10^{-9}	10^{-6}	10^{-3}	10^{-2}	10^{-1}
multiples	symbol	`da`	`h`	`k`	`M`	`G`	`T`
	value	10^{1}	10^{2}	10^{3}	10^{6}	10^{9}	10^{12}

To facilitate explanation, unit operations are detailed below using only scalar values . However, when working with vectors or matrices, the unit operation is performed only once for the entire structure applying the same rules for scalar values, as all numerical elements within any given vector or matrix must have the same physical dimension.

2.4.3.1. Conversion to SI base units

The operation to convert any generic unit to its SI representation involves multiplying its numerical value, χ, by the conversion coefficient γ_{SI}:

$$\chi_{\mathrm{SI}} = \gamma_{\mathrm{SI}}\,\chi \ . \tag{2.13}$$

The conversion coefficient γ_{SI} for the generic unit of measurement is determined by calculating the product of the elements in the coefficient array elevated to the power of the corresponding element in the exponents array:

$$\gamma_{\mathrm{SI}} = \prod_{k=1}^{7} c_k^{\,e_k} \ . \tag{2.14}$$

It should be noted that if the unit is already in its SI representation, the conversion coefficient γ_{SI} will be unitary, and thus no conversion is performed on the numerical value.

As an example, consider the conversion of $\mathrm{cm\,s^{-1}}$ to its SI representation, $\mathrm{m\,s^{-1}}$. The exponent and coefficient arrays related to $\mathrm{cm\,s^{-1}}$ are shown in Equations (2.11) and (2.12). In this specific case, the conversion coefficient is calculated as:

$$\gamma_{\mathrm{SI}} = 1^0 \cdot 0.01^1 \cdot 1^{-1} \cdot 1^0 \cdot 1^0 \cdot 1^0 \cdot 1^0 = 0.01 \ . \tag{2.15}$$

Therefore, the conversion applied to a generic value χ defined in $\mathrm{cm\,s^{-1}}$ to obtain its SI representation is:

$$\chi\ \mathrm{cm\,s^{-1}} = 0.01\chi\ \mathrm{m\,s^{-1}}. \tag{2.16}$$

After applying the conversion coefficient to the numerical value, the elements of the coefficient array $\boldsymbol{c}$ are subsequently set to 1, while the exponent array $\boldsymbol{e}$ remains unchanged.

2.4.3.2. Algebraic Sum

The algebraic sum (addition or subtraction) between two dimensioned values is allowed only when both values represent the same physical quantity; that is, their exponent arrays ($\boldsymbol{e}$) must be identical. Before the operation is executed, the exponent arrays of each value are compared, and if they are not equal, an error message is returned and the process is halted.

For example, if an algebraic sum between a mass (expressed in `Kg`) and a length (expressed

in `m`) is attempted, the following error message will be returned:

```
UnitsArithmeticError: Units [Kg] and [m] are not compatible
                      for the operation: + .
```

When the two dimensioned values share the same exponent array, the operation is performed, and the numerical output, χ_3, is evaluated as:

$$\chi_3 = \gamma_1 \chi_1 + \gamma_2 \chi_2 \,. \tag{2.17}$$

If the two numerical values share the same coefficients array $\boldsymbol{c}$, the conversion factors γ_1 and γ_2 are set to 1, and the output will be provided using the common coefficient. In all other cases, γ_1 and γ_2 are evaluated using Equation (2.14), and the output will be provided in its SI representation. The resultant unit of measurement will have the same exponent array as the two operands, and the elements of the coefficient array ($\boldsymbol{c}$) are all set to 1, since the output is in its SI representation.

2.4.3.3. Multiplication and Division

In contrast to the algebraic sum, it is always possible to multiply dimensioned quantities, provided the correct propagation of the units is ensured.

For multiplication, the numerical output, χ_3, is calculated using the SI conversion factor γ applied to the product of the two operands (χ_1 and χ_2):

$$\chi_3 = \gamma \chi_1 \cdot \chi_2 \,, \tag{2.18}$$

The conversion factor γ is calculated based on the operands' coefficient arrays ($\boldsymbol{c}_1$ and $\boldsymbol{c}_2$) and their individual SI conversion coefficients (γ_1 and γ_2):

$$\gamma = \begin{cases} 1 & \text{if } \boldsymbol{c}_1 = \boldsymbol{c}_2 \\ \gamma_1 \cdot \gamma_2 & \text{if } \boldsymbol{c}_1 \neq \boldsymbol{c}_2 \end{cases} . \tag{2.19}$$

In the first case, where the coefficients are equal, the resulting value will retain that original unit coefficient. In all other cases, the output is converted to its SI representation, and all elements of the resulting coefficient array ($\boldsymbol{c}_3$) are set to 1. The resultant exponent array, $\boldsymbol{e}_3$, is always calculated as the sum of the operands' exponent arrays:

$$\boldsymbol{e}_3 = \boldsymbol{e}_1 + \boldsymbol{e}_2 \,. \tag{2.20}$$

Division follows a parallel logic. The numerical output χ_3 is determined by the division

of the first numerical value with the second, scaled by the factor γ:

$$\chi_3 = \gamma \frac{\chi_1}{\chi_2}\,, \tag{2.21}$$

where γ is calculated as:

$$\gamma = \begin{cases} 1 & \text{if } \boldsymbol{c}_1 = \boldsymbol{c}_2 \\ \dfrac{\gamma_1}{\gamma_2} & \text{if } \boldsymbol{c}_1 \neq \boldsymbol{c}_2 \end{cases}. \tag{2.22}$$

The resultant exponent array $\boldsymbol{e}_3$ is calculated as the difference between the dividend's exponent array ($\boldsymbol{e}_1$) and the divisor's exponent array ($\boldsymbol{e}_2$):

$$\boldsymbol{e}_3 = \boldsymbol{e}_1 - \boldsymbol{e}_2\,. \tag{2.23}$$

As in the case of multiplication, if $\boldsymbol{c}_1 = \boldsymbol{c}_2$, the output dimensioned value will retain the same coefficients as the operands; otherwise, the coefficient array has all elements set to 1, as the output is converted to SI base units.

2.4.3.4. Power

The power operation does not require any prior conversion factor or unit consistency check, as it involves a single dimensioned value. When the dimensioned value χ_1 is raised to the power of ξ, the numerical output χ_2 is simply:

$$\chi_2 = \chi_1^{\xi}\,, \tag{2.24}$$

The resultant exponent array ($\boldsymbol{e}_2$) related to the unit of measurement of χ_2 is calculated by multiplying the original array ($\boldsymbol{e}_1$) by ξ:

$$\boldsymbol{e}_2 = \boldsymbol{e}_1 \xi\,. \tag{2.25}$$

The coefficient array of the output ($\boldsymbol{c}_2$), instead, is obtained by raising $\boldsymbol{c}_1$ to the power of ξ:

$$\boldsymbol{c}_2 = \boldsymbol{c}_1^{\xi}\,. \tag{2.26}$$

2.5. Generalised modelling of source and sink terms

Source and sink terms appear in most of the governing equations describing the physical systems modelled with the present toolbox. Examples include the input of mechanical energy from a pump, kinetic energy losses due to viscosity, or heat generation from electrochemical reactions inside batteries. Within the framework of this library, source and sink terms can be assigned to both nodes and branches using the unique `key` associated with that element. The contribution of a source or sink term is uniformly modelled by a linear polynomial of

the form:

$$\sigma_r = m_r \varphi_r + k_r \,, \tag{2.27}$$

where σ_r is the source term assigned to the node or branch with internal index r, and m_r and k_r are the linear coefficient and the known term, respectively. This formulation is used for any generic governing equation that solves for the field φ_r. This structure also allows for the modelling of non-linear source terms by including the non-linearities in the known term k_r, using the value of the field φ_r from the previous time-step, and subsequently finding σ_r through iterative procedures.

As an example, consider Newton's law of cooling used to model convective heat transfer phenomena, where a fluid at temperature T exchanges heat with a wall considered at constant temperature T_∞. When applied to the generic entity (node or branch) with index j, the source takes the form:

$$\sigma_j = m_j T_j + k_j = \dot{Q} = U_j S_{\text{ref,j}} \left(T_\infty - T_j\right) \,. \tag{2.28}$$

By splitting the terms that multiply the linear term of the temperature field from all other terms, the coefficients m_j and k_j are identified as:

$$m_j = -U_j S_{\text{ref,j}} \,, \tag{2.29}$$

$$k_j = U_j S_{\text{ref,j}} T_\infty \,. \tag{2.30}$$

The same methodology is applied to parametrise all source terms defined within the present toolbox. It is therefore clear that when a source is constant (and hence independent of the field being solved), the coefficient k is simply equal to the value of the constant source:

$$m_j = 0 \,, \quad k_j = \sigma_j \quad \text{if } \sigma_j \text{ is constant} \,. \tag{2.31}$$

2.6. Development of a one-dimensional flow solver

2.6.1. Governing equations

This section analyses the mathematical approach used to develop the fluid flow solver. The model is based on the hypothesis of a single-phase flow characterised by a low Mach number. This means the fluid velocity ($\vec{v}$) is significantly lower than the speed of sound (e.g., 1480 m/s for water and 340 m/s for air at 20°C and ambient pressure). The low Mach number hypothesis permits the fluid to be considered incompressible or quasi-incompressible. Density variations (ρ) due to changes in temperature and pressure are accounted for using an equation of state or by interpolating tabulated data. The governing equations are then solved based on this imposed density field. This approach establishes a two-way coupling between the velocity field ($\vec{v}$) and the temperature field (T), as the velocity depends on

density (which depends on temperature and pressure), and the temperature field itself depends on the velocity. Under the mentioned hypotheses, the set of governing equations is composed of the macroscopic balances of mass, mechanical energy, and thermal energy. These equations are presented below in their integral form, applied over a Control Volume (CV) V bounded by the surface S.

The macroscopic balance of mass is given by:

$$\int_V \frac{\partial \rho}{\partial t}\, dV + \int_S \rho\, \vec{v} \cdot \vec{n}\, dS = 0\,, \tag{2.32}$$

where the first term represents the time rate of change of mass within the volume, and the second term is the net mass flux across the surface S (convective term).

The macroscopic balance of mechanical energy is expressed as:

$$\begin{aligned} \frac{d}{dt}\int_{V(t)} \rho \left(\frac{v^2}{2} + g\,z\right) dV = &- \int_S \rho \left(\frac{v^2}{2} + g\,z\right)(\vec{v} - \vec{v}_s)\cdot \vec{n}\, dS \\ &- \int_S p\, \vec{v}\cdot\vec{n}\, dS \; + \int_S (\boldsymbol{\tau}\vec{v})\cdot\vec{n}\, dS \\ &+ \int_V p \nabla\cdot\vec{v}\, dV - \int_V \boldsymbol{\tau} : \nabla\vec{v}\, dV\,, \end{aligned} \tag{2.33}$$

This equation is derived by multiplying the mass balance (Equation (2.32)) by the velocity field $\vec{v}$ and employing the Leibnitz formula for integrating over a moving CV $V(t)$, where $\vec{v}_s$ is the velocity of the CV surface. The terms represent: time rate of change of mechanical energy in the Left Hand Side (LHS), convective transport, work done by pressure forces, work done by viscous shear stress ($\boldsymbol{\tau}$), work due to volume dilation, and viscous dissipation, respectively.

The macroscopic balance of thermal energy is given by:

$$\begin{aligned} \frac{\partial}{\partial t}\int_V \rho\, c_p\, T\, dV + \int_S \rho\, c_p\, T\, \vec{v}\cdot\vec{n}\, dS = &+ \int_S -\vec{q}\cdot\vec{n}\, dS + \int_V H\, dV \\ &+ \int_V \boldsymbol{\tau} : \nabla\vec{v}\, dV\,. \end{aligned} \tag{2.34}$$

The terms in this equation represent: time rate of change of internal energy (LHS, first term), convective transport of enthalpy (LHS, second term), heat transfer trough the surface of the branch CV ($\vec{q}$), volumetric heat source or sink (H), and conversion of mechanical energy into heat via viscous dissipation ($\boldsymbol{\tau} : \nabla\vec{v}$), respectively.

By numerically solving the governing equation, the solver provides the resulted field of velocity, temperature, density and pressure.

2.6.2. Discretisation schemes

The solution of the governing equations is based on the domain discretisation schemes. In fact, the domain, which is described as a graph of nodes and branches, is discretised using two simultaneous schemes: one for node CVs and one for branch CVs.

- Node Control Volume: Node CVs include the node itself and half of each branch connected to it. The boundaries of the control volume are placed at the centre of each connected branch and are considered planar and normal to the branch axes. Figure 2.2a shows the control volume associated with a node connected to three branches. The normal unit vector in the boundaries are considered positive if pointing outwards. In the figure the blue points are the nodes and the red points are the centre of the branches.

- Branch control volume: Branch CVs include the fluid volume represented by the branch itself, considering an uniform and arbitrary cross section across the length. The boundaries of each CV are in correspondence of the starting and ending nodes, and they are considered planar and perpendicular to the axis of the represented branch. In Figure 2.2b is shown the control volume of a branch with circular cross section on which are defined also an heat source, represented by the resistance, and a mechanical energy source, represented by the pump. From the figure is possible to note that the source elements are external to the control volume and is considered only the overall effects of these sources or sinks on the system.

The mass balance and thermal energy balance are solved in the node CVs, instead the mechanical energy balance is solved in the branch CVs. As shown in Figure 2.3, the temperature, density and pressure (T, ρ, P) are defined to the nodes, while the velocity (v) is defined on the branches and is considered positive when has the same direction of the branch. Since in the mechanical energy balance enters also variables defined on the nodes (such as P and ρ), it is necessary to include an operator which transfers the data from the nodes to the branches. This is achieved by using the first order upwind scheme, explained in more detail in Appendix A.1, to find the operator $\boldsymbol{B}$ which multiplies the nodal value φ_{n} and calculate its corresponding branch value φ_{b}:

$$\varphi_{\mathrm{b}} = \boldsymbol{B}\,\varphi_{\mathrm{n}}\,, \tag{2.35}$$

The operator $\boldsymbol{B}$ is formulated as:

$$\boldsymbol{B} = 0.5\left(\left|\boldsymbol{A}^{\mathrm{T}}\right| + \boldsymbol{O}\boldsymbol{A}^{\mathrm{T}}\right)\,. \tag{2.36}$$

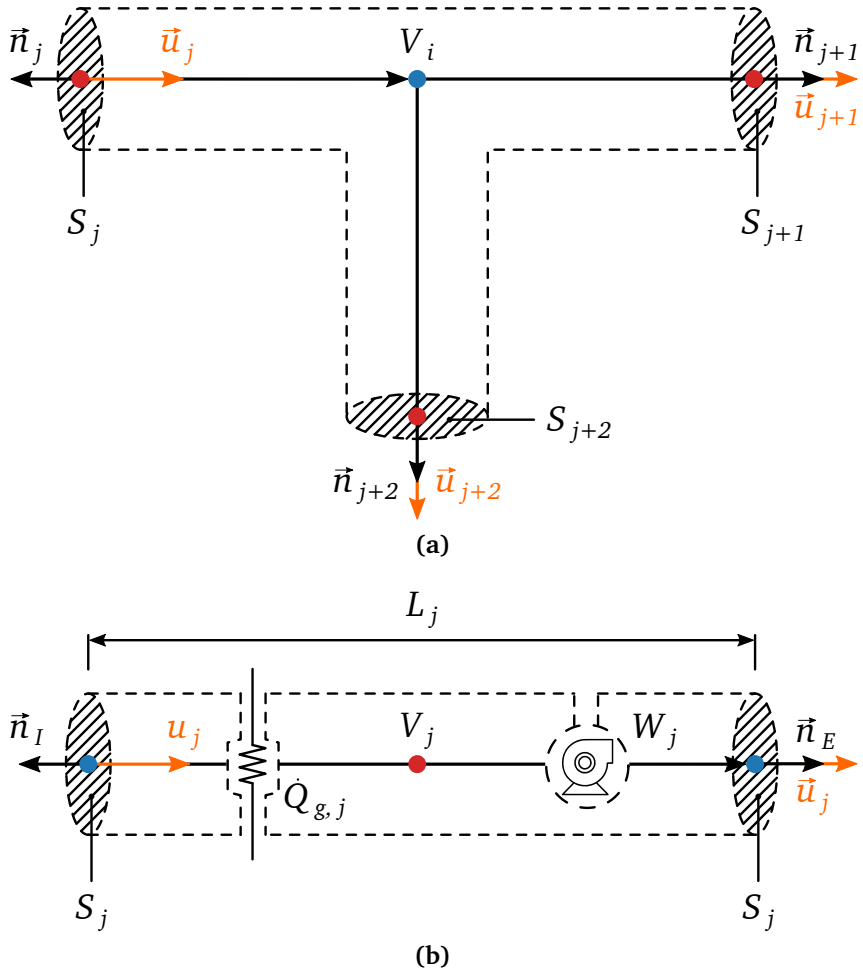


Figure 2.2: Scheme of a generic node control volume (a) and branch control volume (b), where blue points stand for nodes while the red points are the centres of branches. Images adapted from [132].

In Equation (2.36), $\boldsymbol{O}$ is an $n_\mathrm{b} \times n_\mathrm{b}$ diagonal matrix which contains the orientations of branch velocities:

$$O_{jj} = \frac{u_j}{|u_j|} \,. \tag{2.37}$$

The upwind scheme operator $\boldsymbol{B}$ is essentially a modified version of the transposed incidence matrix: for any given row, positive elements are kept if the fluid velocity of the related branch is positive (*i.e.* has the same direction of the branch), meaning that the upwind node, which is the node where the flow originates, is the starting point of the branch considered; instead, negative elements are deleted. When the fluid velocity is negative, positive elements are deleted, while negative ones are kept in absolute value, meaning that the upwind node is the end point of the branch under examination. This mechanism ensures that the nodal value from the upwind node is assigned to the generic branch field.

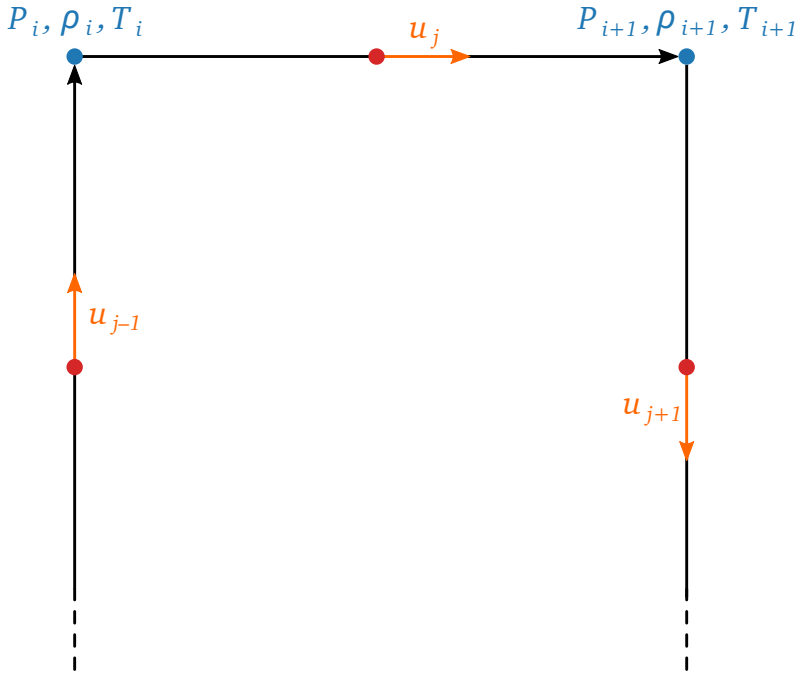


Figure 2.3: Example of staggered arrangement of nodes and branches; blue points stand for nodes, while red points represent the centre of branches. Image adapted from [132].

2.6.2.1. Macroscopic mass balance

At each node of the domain is ensured the mass conversation, where the mass flows through the inlet/outlet ports placed at the boundaries of node CVs, at the centre of each connected branch. The mass flow velocity, in each boundary, is assumed aligned with the branch direction, since the vents are assumed planar and fixed in space. To numerically solve the governing equations, they need to be re-arranged in an algebraic equation. To do this the integrals in Equation (2.32) are solved by means of the second order midpoint method, which substitute the integral averages of density and velocity with their midpoint values; for a more detailed explanation, refer to Appendix A.2. For the generic node with index i, the mass conservation reads:

$$\frac{\partial \bar{\rho}_i}{\partial t} V_i + \sum_j^{n_E} \bar{\rho}_j \bar{u}_j S_j - \sum_j^{n_I} \bar{\rho}_j \bar{u}_j S_j = 0 \,, \tag{2.38}$$

where u is the branch-wise velocity component, S_j is the cross-sectional area of the branch, and, n_I and n_E are the number of branches entering and exiting the node i. The quantities with over-bars are taken at the middle point of the volume or surface considered. To integrate the equation in the time-step $\Delta t = t^{n+1} - t^n$, Equation (2.38) is re-arranged in the form:

$$\left(\bar{\rho}_i^{n+1} - \bar{\rho}_i^n\right) V_i = \int_n^{n+1} \left(\sum_j^{n_I} \bar{\rho}_j \bar{u}_j S_j - \sum_j^{n_E} \bar{\rho}_j \bar{u}_j S_j \right) dt \,. \tag{2.39}$$

By employing the first order implicit Euler method (derived in Appendix A.3), the time integral in the Right Hand Side (RHS) is solved, and the equation becomes:

$$\frac{\bar{\rho}_i^{\,n+1} - \bar{\rho}_i^{\,n}}{\Delta t} V_i = \sum_j^{n_\mathrm{I}} \bar{\rho}_j^{\,n+1} \bar{u}_j^{\,n+1} S_j - \sum_j^{n_\mathrm{E}} \bar{\rho}_j^{\,n+1} \bar{u}_j^{\,n+1} S_j \,. \tag{2.40}$$

The sign of mass fluxes in the RHS of Equation (2.40) is then managed by means of the incidence matrix. In fact, according to Equation (2.5), when a branch comes out from the node i are associated to the value of 1, while the branches entering the node are associated to the value of -1. Therefore, using the incidence matrix, Equation (2.40) is rewritten as:

$$\sum_{j=1}^{n_\mathrm{b}} A_{ij} \bar{\rho}_j^{\,n+1} \bar{u}_j^{\,n+1} S_j = -\frac{\bar{\rho}_i^{\,n+1} - \bar{\rho}_i^{\,n}}{\Delta t} V_i \,. \tag{2.41}$$

Since Equation (2.41) is solved for each node, it is possible to create the system of n_n equations

$$\boldsymbol{A}\boldsymbol{M}\,\boldsymbol{u}^{\,n+1} = -\boldsymbol{d} \,, \tag{2.42}$$

where $\boldsymbol{M}$ is a $n_\mathrm{b} \times n_\mathrm{b}$ diagonal matrix of elements:

$$M_{jj} = \bar{\rho}_j^{\,n+1} S_j \,, \tag{2.43}$$

while $\boldsymbol{u}^{\,n+1}$ is the array of branch velocities at the time-step $n+1$. The term $\boldsymbol{d}$ is an array containing all the unsteady terms:

$$d_i = \frac{\bar{\rho}_i^{\,n+1} - \bar{\rho}_i^{\,n}}{\Delta t} V_i \,. \tag{2.44}$$

In steady-state problems, the unsteady terms are neglected, therefore the system of equations becomes:

$$\boldsymbol{A}\boldsymbol{M}\,\boldsymbol{u} = \boldsymbol{0} \,. \tag{2.45}$$

2.6.2.2. Macroscopic mechanical energy balance

As for the the mass conservation equation, here is derived an algebraic equation for the mechanical energy balance. Starting from its integral form, the volume integral in the LHS of Equation (2.33) is solved by means of the midpoint rule:

$$\frac{d}{dt}\int_{V(t)} \rho \left(\frac{v^2}{2} + g\,z\right) dV \approx \frac{d}{dt}\left[\rho_j \left(\frac{\bar{u}_j^2}{2} + g\,\bar{z}_j\right) V_j\right] . \tag{2.46}$$

Since the mechanical energy balance is solved in the branch CVs, V_j, in Equation (2.46), is the volume of the branch, that is equal to its cross-section multiplied for the total length:

$$V_j = S_j L_j \tag{2.47}$$

The advective terms, *i.e.*, the first and second term in the RHS of Equation (2.33), are also solved using the midpoint rule:

$$\begin{aligned}\int_S \rho \left(\frac{v^2}{2} + g\,z\right)(\vec{v} - \vec{v}_s)\cdot\vec{n}\,dS \approx \bar{\rho}_{\mathrm{E}}\left(\frac{\bar{u}_j^{\,2}}{2} + g\,\bar{z}_{\mathrm{E}}\right)\bar{u}_j S_j \\ - \bar{\rho}_{\mathrm{I}}\left(\frac{\bar{u}_j^{\,2}}{2} + g\,\bar{z}_{\mathrm{I}}\right)\bar{u}_j S_j\,,\end{aligned} \tag{2.48}$$

$$\int_S p\,\vec{v}\cdot\vec{n}\,dS \approx \bar{p}_{\mathrm{E}}\,\bar{u}_j S_j - \bar{p}_{\mathrm{I}}\,\bar{u}_j S_j - \dot{W}_{\mathrm{p},j}\,, \tag{2.49}$$

where $\dot{W}_{\mathrm{p},j}$ is the work per unit of time of the fluid on impermeable walls. The third term in the RHS of Equation (2.33), is the work per unit of time from the viscous forces:

$$\int_S (\boldsymbol{\tau}\,\vec{v})\cdot\vec{n}\,dS = \dot{W}_{\tau,j}\,. \tag{2.50}$$

The fourth therm in the RHS of Equation (2.33) is the compression power. This term usually is neglected in the mechanical energy balance of lumped parameter solvers, since its effects are accounted for in other ways, like the work performed or extracted by fluid machines and head losses. Finally the last term of Equation (2.33) represents the power dissipated due to viscous effects:

$$\int_V \boldsymbol{\tau} : \nabla\vec{v}\,dV = \dot{E}_{v,j}\,. \tag{2.51}$$

At this point each term of Equation (2.33), can be substituted with its corresponding algebraic version (Equations (2.46) to (2.51)), obtaining:

$$\begin{aligned}\frac{d}{dt}\left[\rho_j\left(\frac{\bar{u}_j^{\,2}}{2} + g\,\bar{z}_j\right)V_j\right] = -\left(\bar{\rho}_{\mathrm{E}}\frac{\bar{u}_j^{\,2}}{2} + \bar{p}_{\mathrm{E}} + \bar{\rho}_{\mathrm{E}}\,g\,\bar{z}_{\mathrm{E}}\right)\bar{u}_j S_j \\ + \left(\bar{\rho}_{\mathrm{I}}\frac{\bar{u}_j^{\,2}}{2} + \bar{p}_{\mathrm{I}} + \bar{\rho}_{\mathrm{I}}\,g\,\bar{z}_{\mathrm{I}}\right)\bar{u}_j S_j \\ + \dot{W}_{\mathrm{p},j} + \dot{W}_{\tau,j} - \dot{E}_{v,j}\,.\end{aligned} \tag{2.52}$$

It is defined the total pressure of the fluid at the generic i-th node as the sum of the dynamic pressure, static pressure and hydrostatic pressure:

$$P_i = \bar{\rho}_i \frac{\bar{u}_j^2}{2} + \bar{p}_i + \bar{\rho}_i g \bar{z}_i \,. \tag{2.53}$$

In addition the gravitational energy time derivative can be neglected since branch position does not change in time, therefore Equation (2.52) simplifies to:

$$\frac{d}{dt}\left(\bar{\rho}_j \frac{\bar{u}_j^2}{2}\right) V_j = P_\mathrm{I} \bar{u}_j S_j - P_\mathrm{E} \bar{u}_j S_j + \dot{W}_j - \dot{E}_{v,j} \,, \tag{2.54}$$

$$\dot{W}_j = \dot{W}_{\mathrm{p},j} + \dot{W}_{\tau,j} \,, \tag{2.55}$$

where $\dot{W}_j$ is the external work per unit of time done or extract by mechanical devices such as fans or, in the case of Figure 2.2b, pumps. The time derivative in Equation (2.54) is solved by employing the Implicit Euler method, hence the mechanical energy balance in its algebraic form is:

$$\begin{aligned} \frac{V_j}{\Delta t}\left[\bar{\rho}_j^{n+1} \frac{\left(\bar{u}_j^{n+1}\right)^2}{2} - \bar{\rho}_j^{n} \frac{\left(\bar{u}_j^{n}\right)^2}{2}\right] = &+ P_\mathrm{I}^{n+1} \bar{u}_j^{n+1} S_j \\ &- P_\mathrm{E}^{n+1} \bar{u}_j^{n+1} S_j \\ &+ \dot{W}_j^{n+1} - \dot{E}_{v,j}^{n+1} \end{aligned} \tag{2.56}$$

By dividing all the terms by $\bar{u}_j^{n+1} S_j$, which is the volumetric flow rate of the branch j at the time-step $n+1$, the macroscopic mechanical energy balance reads:

$$\begin{aligned} \frac{L_j}{\Delta t}\left[\bar{\rho}_j^{n+1} \frac{\bar{u}_j^{n+1}}{2} - \bar{\rho}_j^{n} \frac{\left(\bar{u}_j^{n}\right)^2}{2 u_j^{n+1}}\right] = &P_\mathrm{I}^{n+1} - P_\mathrm{E}^{n+1} \\ &+ \Delta p_{\mathrm{P},j}^{n+1} - \Delta p_{\mathrm{L},j}^{n+1} \,, \end{aligned} \tag{2.57}$$

where $\Delta p_{\mathrm{P},j}^{n+1}$ is the pressure difference caused by the presence of pumps or fans, and $\Delta p_{\mathrm{L},j}^{n+1}$ represents the pressure losses due to viscous effects. The effect of this terms are considered as momentum sources, so they become generalised source as described in Equation (2.27), obtaining:

$$\begin{aligned} \frac{L_j}{\Delta t}\left[\bar{\rho}_j^{n+1} \frac{\bar{u}_j^{n+1}}{2} - \bar{\rho}_j^{n} \frac{\bar{u}_j^{n}}{2}\right] = &+ P_\mathrm{I}^{n+1} - P_\mathrm{E}^{n+1} \\ &+ \sigma_{\mathrm{P},j}^{n+1} - \sigma_{\mathrm{L},j}^{n+1} \,, \end{aligned} \tag{2.58}$$

in the above equation an error is introduced, since it is assumed

$$\frac{\left(\bar{u}_j^n\right)^2}{2u_j^{n+1}} = \frac{\bar{u}_j^n}{2}\,, \tag{2.59}$$

the error is a first-order error, in fact:

$$e = \frac{(u^n)^2}{u^n} - \frac{(u^n)^2}{u^{n+1}} = \frac{du}{dt}\Delta t - \left(\frac{du}{dt}\right)^2 \frac{\Delta t^2}{u^{n+1}} + \ldots \tag{2.60}$$

Finally Equation (2.56) is reformulated by introducing the generalised source coefficients m and k and by reordering the terms:

$$\begin{aligned} P_{\mathrm{I}}^{n+1} - P_{\mathrm{E}}^{n+1} = &\left(m_{\mathrm{L},j}^{n+1} - m_{\mathrm{P},j}^{n+1} + \frac{\bar{\rho}_j^{n+1} L_j}{2\Delta t} \right) \bar{u}_j^{n+1} \\ &- \left(k_{\mathrm{P},j}^{n+1} + \frac{\bar{\rho}_j^{n} \bar{u}_j^{n} L_j}{2\Delta t} \right). \end{aligned} \tag{2.61}$$

The above equation is solved for all the branches of the domain, constituting a system of equations:

$$\boldsymbol{A}^{\mathrm{T}} \boldsymbol{P}^{n+1} = \boldsymbol{Y}\,\boldsymbol{u}^{n+1} + \boldsymbol{k}\,, \tag{2.62}$$

where $\boldsymbol{A}$ is the incidence matrix (Equation (2.4)), $\boldsymbol{P}^{n+1}$ is the vector of the total pressure on the nodes at the time-step $n+1$, and $\boldsymbol{Y}$ is an $n_{\mathrm{b}} \times n_{\mathrm{b}}$ diagonal matrix of elements:

$$Y_{jj} = m_{\mathrm{L},j}^{n+1} - m_{\mathrm{P},j}^{n+1} + \frac{\rho_j^{n+1} L_j}{2\Delta t}\,. \tag{2.63}$$

In Equation (2.62) $\boldsymbol{k}$ is the vector of known terms, with elements:

$$k_j = -k_{\mathrm{P},j}^{n+1} - \frac{\bar{\rho}_j^{n} \bar{u}_j^{n} L_j}{2\Delta t}\,. \tag{2.64}$$

For steady-state conditions, the unsteady terms are neglected, hence equations (2.63) and (2.64) become respectively:

$$Y_{jj} = m_{\mathrm{L},j} - m_{\mathrm{P},j} \tag{2.65}$$

$$k_j = -k_{\mathrm{P},j}\,. \tag{2.66}$$

2.6.2.3. Thermal energy macroscopic balance

The same derivation approach used to define the macroscopic mass and mechanical energy balances is applied to the integral thermal energy conservation equation (Equation (2.34)); this equation is solved on each node, together with the mass conservation. By

employing the midpoint rule, the unsteady term and the convective term in the LHS of Equation (2.34) are discretised as:

$$\frac{\partial}{\partial t}\int_V \rho\, c_\mathrm{p}\, T\, dV \approx \frac{\partial}{\partial t}\left(\bar{\rho}_i\, \bar{c}_{p,i}\, \bar{T}_i\right) V_i\,, \tag{2.67}$$

$$\int_S \rho\, c_\mathrm{p}\, T\, \vec{v}\cdot\vec{n}\, dS \approx \sum_j^{n_\mathrm{E}} C_{\mathrm{r},j}\, T_j - \sum_j^{n_\mathrm{I}} C_{\mathrm{r},j}\, T_j\,, \tag{2.68}$$

where $C_{\mathrm{r},j}$ is the heat capacity rate related to branch j. The first and second terms in the RHS of Equation (2.34), which define the heat transferred through the surface of branch j and the volumetric heat generation, respectively, are treated as generalised source or sink terms:

$$\int_S -\vec{q}\cdot\vec{n}\, dS = m_{\mathrm{e},j}\, \bar{T}_j + k_{\mathrm{e},j}\,, \tag{2.69}$$

$$\int_V H\, dV = m_{\mathrm{e},j}\, \bar{T}_j + k_{\mathrm{e},j}\,. \tag{2.70}$$

In addition heat diffusion along the branchwise direction is neglected. By assembling the terms from Equation (2.67) to Equation (2.70), and including the viscous dissipation term, the thermal energy balance is obtained:

$$\begin{aligned}&\frac{\partial}{\partial t}\left(\bar{\rho}_i\, \bar{c}_{p,i}\, \bar{T}_i\right) V_i + \sum_j^{n_\mathrm{E}} C_{\mathrm{r},j}\, T_j - \sum_j^{n_\mathrm{I}} C_{\mathrm{r},j}\, T_j \\ &+ \sum_j^{n_\mathrm{I}}\left(-m_{\mathrm{e},j}\right) \bar{T}_j = \sum_j^{n_\mathrm{I}}\left(k_{\mathrm{e},j} + \Delta p_{\mathrm{L},j}\, u_j\, S_j\right)\,.\end{aligned} \tag{2.71}$$

The time derivative in the above equation is numerically solved with the Implicit Euler method; therefore, the macroscopic thermal energy balance is defined as:

$$\begin{aligned}&\frac{V_i}{\Delta t}\left(\bar{\rho}_i^{\,n+1}\, \bar{c}_{\mathrm{p},i}^{\,n+1}\, \bar{T}_i^{\,n+1} - \bar{\rho}_i^{\,n}\, \bar{c}_{\mathrm{p},i}^{\,n}\, \bar{T}_i^{\,n}\right) + \sum_j^{n_\mathrm{E}} C_{\mathrm{r},j}^{n+1}\, \bar{T}_j^{\,n+1} \\ &- \sum_j^{n_\mathrm{I}} C_{\mathrm{r},j}^{n+1}\, \bar{T}_j^{\,n+1} + \sum_j^{n_\mathrm{I}}\left(-m_{\mathrm{e},j}^{n+1}\right) \bar{T}_j^{\,n+1} = \sum_j^{n_\mathrm{I}}\left(k_{\mathrm{e},j}^{n+1} + \Delta p_{\mathrm{L},j}^{n+1}\, u_j^{n+1}\, S_j\right)\,.\end{aligned} \tag{2.72}$$

Introducing the incidence matrix $\boldsymbol{A}$ in Equation (2.72) makes it possible to manage the sign of the fluxes through the inlet and outlet ports. As shown in Equation (2.5), the incidence matrix assigns a coefficient of -1 to branches entering the node i and a coefficient of $+1$ to branches exiting the node; hence, the macroscopic energy balance can be expressed as

follows:

$$\begin{aligned}&\frac{V_i}{\Delta t}\left(\bar{\rho}_i^{\,n+1}\,\bar{c}_{\mathrm{p},i}^{\,n+1}\,\bar{T}_i^{\,n+1}-\bar{\rho}_i^{\,n}\,\bar{c}_{\mathrm{p},i}^{\,n}\,\bar{T}_i^{\,n}\right)+\sum_j^{n_\mathrm{b}}A_{ij}\,C_{\mathrm{r},j}^{n+1}\,\bar{T}_j^{\,n+1}\\&+\sum_j^{n_\mathrm{I}}\left(-m_{\mathrm{e},j}^{n+1}\right)\bar{T}_j^{\,n+1}=\sum_j^{n_\mathrm{I}}\left(k_{\mathrm{e},j}^{n+1}+\Delta p_{\mathrm{L},j}^{n+1}\,u_j^{n+1}\,S_j\right).\end{aligned}\tag{2.73}$$

To manage the summation of the $-m_{\mathrm{e},j}^{n+1}$ and $k_{\mathrm{e},j}^{n+1}$ terms over inlet branches, the matrix operator $\boldsymbol{N}$ is introduced, which is based on the incidence matrix:

$$\boldsymbol{N}=0.5\,(|\boldsymbol{A}|-\boldsymbol{A}\,\boldsymbol{O})\,,\tag{2.74}$$

where $\boldsymbol{O}$ is defined in Equation (2.37). Therefore, when the velocity of branch j is positive, the column of $\boldsymbol{N}$ referring to branch j keeps the negative element of the same column in the incidence matrix and discards the positive one; hence, the downstream node is the ending node of the branch. In the opposite case, when the branch velocity is negative, only the positive coefficients are kept. Equation (2.73) becomes:

$$\begin{aligned}&\frac{V_i}{\Delta t}\bar{\rho}_i^{\,n+1}\,\bar{c}_{\mathrm{p},i}^{\,n+1}\,\bar{T}_i^{\,n+1}+\sum_j^{n_\mathrm{b}}A_{ij}\,C_{\mathrm{r},j}^{n+1}\,\bar{T}_j^{\,n+1}+\sum_j^{n_\mathrm{b}}N_{ij}\left(-m_{\mathrm{e},j}^{n+1}\right)\bar{T}_j^{\,n+1}=\\&=\sum_j^{n_\mathrm{b}}N_{ij}\left(k_{\mathrm{e},j}^{n+1}+\Delta p_{\mathrm{L},j}^{n+1}\,u_j^{n+1}\,S_j\right)+\frac{V_i}{\Delta t}\,\bar{\rho}_i^{\,n}\,\bar{c}_{\mathrm{p},i}^{\,n}\,\bar{T}_i^{\,n}\,.\end{aligned}\tag{2.75}$$

To evaluate the temperature related to the generic branch j, the upwind operator $\boldsymbol{B}$ is employed:

$$\begin{aligned}&\frac{V_i}{\Delta t}\bar{\rho}_i^{\,n+1}\,\bar{c}_{\mathrm{p},i}^{\,n+1}\,\bar{T}_i^{\,n+1}+\sum_j^{n_\mathrm{b}}A_{ij}\,C_{\mathrm{r},j}^{n+1}\sum_i^{n_\mathrm{n}}B_{ji}\,\bar{T}_i^{\,n+1}+\sum_j^{n_\mathrm{b}}N_{ij}\,(-m_{\mathrm{e},j}^{n+1})\sum_i^{n_\mathrm{n}}B_{ji}\,\bar{T}_i^{\,n+1}\\&=\sum_j^{n_\mathrm{b}}N_{ij}\left(k_{\mathrm{e},j}^{n+1}+\Delta p_{\mathrm{L},j}^{n+1}\,u_j^{n+1}\,S_j\right)+\frac{V_i}{\Delta t}\,\bar{\rho}_i^{\,n}\,\bar{c}_{\mathrm{p},i}^{\,n}\,\bar{T}_i^{\,n}\,.\end{aligned}\tag{2.76}$$

Applying Equation (2.76) to each node CV, the system of n_n equations is assembled in a compact matrix form:

$$\boldsymbol{W}\,\boldsymbol{T}^{\,n+1}=\boldsymbol{s}\,,\tag{2.77}$$

with the coefficient matrix $\boldsymbol{W}$ defined as:

$$\boldsymbol{W}=\boldsymbol{W}_\mathrm{u}+(\boldsymbol{A}\,\boldsymbol{C}+\boldsymbol{N}\,\boldsymbol{M}_\mathrm{e})\,\boldsymbol{B}\,,\tag{2.78}$$

In Equation (2.78), $\boldsymbol{C}$ is an $n_\mathrm{b} \times n_\mathrm{b}$ diagonal matrix containing the heat capacity rate of the branches, $\boldsymbol{M}_\mathrm{e}$ is also an $n_\mathrm{b} \times n_\mathrm{b}$ diagonal matrix composed of the $-m_{\mathrm{e},j}^{n+1}$ coefficients, and $\boldsymbol{W}_\mathrm{u}$ is an $n_\mathrm{n} \times n_\mathrm{n}$ diagonal matrix with the time dependent terms of the time-step $n+1$:

$$W_{\mathrm{u},ii} = \frac{V_i}{\Delta t}\, \bar{\rho}_i^{\,n+1}\, \bar{c}_{\mathrm{p},i}^{\,n+1} \,. \tag{2.79}$$

The elements composing the known term vector $\boldsymbol{s}$ in Equation (2.77) are calculated by:

$$s_i = \sum_j^{n_\mathrm{b}} N_{ij} \left(k_{\mathrm{e},j}^{n+1} + \Delta p_{\mathrm{L},j}^{n+1}\, u_j^{n+1}\, S_j \right) + \frac{V_i}{\Delta t}\, \bar{\rho}_i^{\,n}\, \bar{c}_{\mathrm{p},i}^{\,n}\, \bar{T}_i^{\,n} \,. \tag{2.80}$$

Finally, in steady-state problems the unsteady terms in equations (2.78) and (2.80) are neglected:

$$\boldsymbol{W} = (\boldsymbol{A}\boldsymbol{C} + \boldsymbol{N}\boldsymbol{M}_\mathrm{e})\, \boldsymbol{B} \,, \tag{2.81}$$

$$s_i = \sum_j^{n_\mathrm{b}} N_{ij} \left(k_{\mathrm{e},j}^{n+1} + \Delta p_{\mathrm{L},j}^{n+1}\, u_j^{n+1}\, S_j \right) \,. \tag{2.82}$$

2.6.3. Modelling of components of networks of pipes

2.6.3.1. Centrifugal pumps and fans

In many applications of pipe networks, there are centrifugal machines, like pumps or fans, that are responsible for moving the operating fluid. These machines exploit centrifugal forces to increase the mechanical energy of the fluid; generally, the relation for the pressure head provided by a pump is a quadratic function of the fluid velocity:

$$\Delta p_\mathrm{P} = c_0 + c_1\, u + c_2\, u^2 \,. \tag{2.83}$$

Equation (2.83) is valid for a fixed operational regime, which is identified by the rotational speed of the pump shaft. By exploiting the hydraulic similarity theory [133] it is possible to derive a non-dimensional curve to relate the head, flow rate and speed of the shaft:

$$\psi = c'_0 + c'_1\, \phi + c'_2\, \phi^2 \,. \tag{2.84}$$

In Equation (2.84), the quantities ψ and ϕ represent the head and flow rate coefficients. In addition it is assumed that all the working points, defined by coordinates (ϕ, ψ), lay on the same hydraulic efficiency curve.

$$\phi = \frac{u\, S}{\mathrm{rps}\, D_\mathrm{ref}^3} \,, \tag{2.85}$$

$$\psi = \frac{\Delta p_\mathrm{P}}{\rho\, (\mathrm{rps}\, D_\mathrm{ref})^2} \,. \tag{2.86}$$

In the above equations, D_{ref} is the reference length of the pump, usually is taken the diameter of the impeller as reference, rps is the rotational speed of the shaft in revolutions per second. Figure 2.4 plots in yellow the Equation (2.84). The curve is a downward-facing parabola, the minimum value of the head coefficient ψ is in correspondence of the maximum value of the coefficient ϕ, then moving towards lower values of ϕ the head coefficient increase until its maximum value $\psi_{\text{v}} = 0.88$ in the point at $\phi_{\text{v}} = 0.30$, then for lower values of ϕ the curve decreases. This non-monotonic trend causes convergence issues when, at the first iterations, the flow coefficient has small values. Hence, to improve the stability of the iterative procedure, Equation (2.84) is modified in a smooth monotonic function, by first mirroring the yellow curve in Figure 2.4 with respect to an horizontal axis passing for the point $(\phi_{\text{v}}, \psi_{\text{v}})$ and then combine Equation (2.84) with the equation of the mirrored curve. In Figure 2.4 the mirrored curve is the blue one and its equation is:

$$\psi' = 2\,\psi(\phi_{\text{v}}) - c'_0 - c'_1\,\phi - c'_2\,\phi^2\,. \tag{2.87}$$

By combining Equation (2.84) with Equation (2.87), is obtained the black curve in Figure 2.4, which represents the smooth monotonic function used in the algorithm for modelling centrifugal machines:

$$\psi'' = \begin{cases} 2\,\psi(\phi_{\text{v}}) - c'_0 - c'_1\,\phi - c'_2\,\phi^2 & \text{for } \phi < \phi_{\text{v}} \\ c'_0 + c'_1\,\phi + c'_2\,\phi^2 & \text{for } \phi \geqslant \phi_{\text{v}} \end{cases}. \tag{2.88}$$

In cases when $\phi < \phi_{\text{v}}$, Equations (2.88) overestimate the value of the head coefficient, but if the circuit is correctly designed, the operating point of the pump is located in points with $\phi > \phi_{\text{v}}$, where ψ'' takes the same values of ψ.

The momentum source given by centrifugal machines is expressed in terms of pressure difference; according to Equation (2.27), the generalised momentum source for the j-th branch at the step $k+1$, takes the form:

$$\Delta p_{\text{P},j}^{k+1} \approx \sigma_{Pj}^{k+1} = m_{\text{P},j}\,u_j^{k+1} + k_{Pj}\,, \tag{2.89}$$

therefore it is necessary to linearise Equations (2.88). This is done by using the flow rate coefficient at the previous time-steps:

$$\psi_j{}'' \approx \xi_j\,\phi_j^{\,k} + \gamma_j\,, \tag{2.90}$$

$$\xi_j = \begin{cases} -c'_{1j} - c'_{2j}\,\phi_j^{k-1} & \text{for } \phi_j^{k-1} < \phi_{vj} \\ c'_{1j} + c'_{2j}\,\phi_j^{k-1} & \text{for } \phi_j^{k-1} \geqslant \phi_{vj} \end{cases}, \tag{2.91}$$

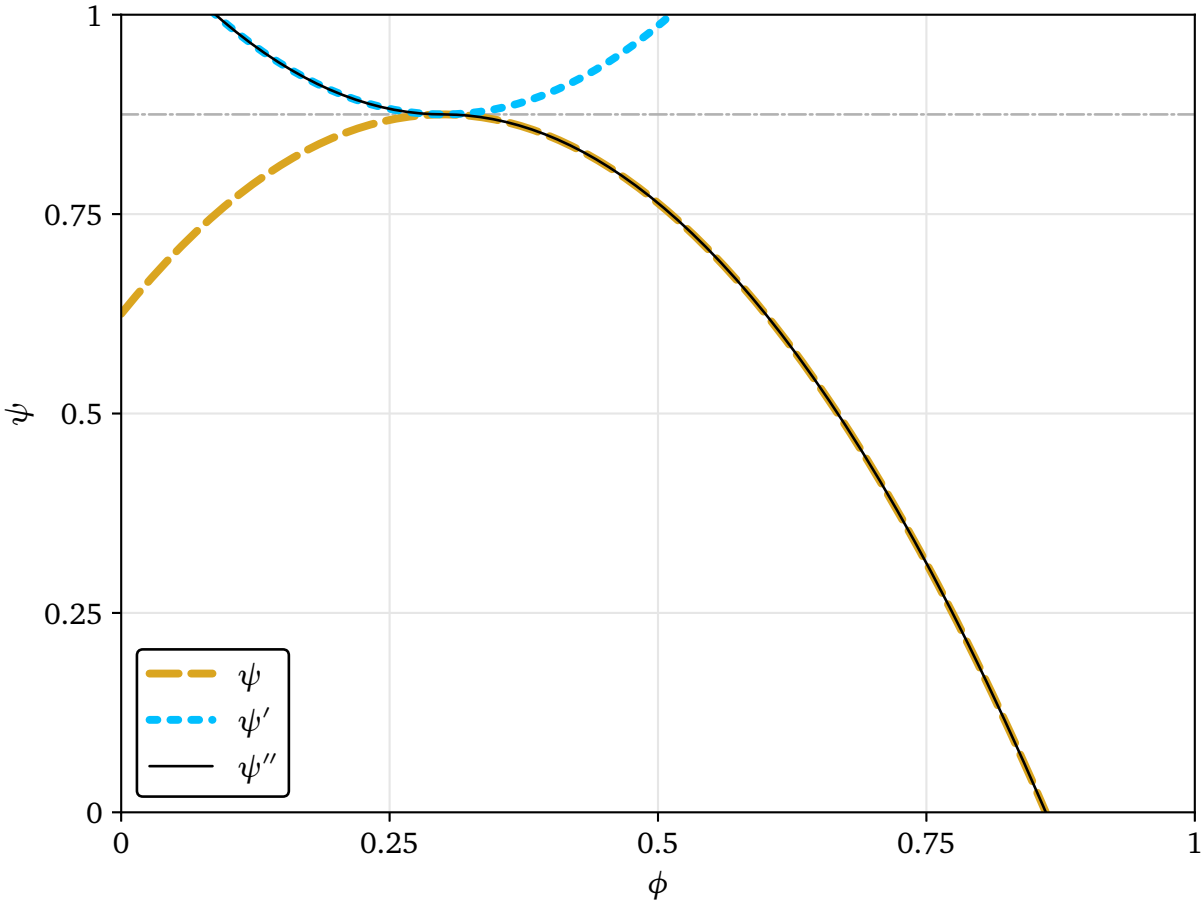


Figure 2.4: Example of a characteristic non-dimensional curves showing the relation between flow rate and head coefficient. The yellow curve is the real curve, while the black line represents the curve used for modelling centrifugal pumps and fans. Image from [132]

$$\gamma_j = \begin{cases} 2\psi_j(\phi_{vj}) - c'_{0j} & \text{for } \phi_j^{k-1} < \phi_{vj} \\ c'_{0j} & \text{for } \phi_j^{k-1} \geqslant \phi_{vj} \end{cases} . \tag{2.92}$$

The dimensional version of Equation (2.90) is obtained by multiplying all the members by $\rho_j (rps_j D_{ref\,j})^2$:

$$\Delta p_{P\,j}^k \approx \sigma_{P\,j}^k = \frac{\rho_j\, rps_j\, S_j}{D_{ref\,j}} \xi_j u_j^k + \rho_j (rps_j D_{ref\,j})^2 \gamma_j . \tag{2.93}$$

Finally, from Equation (2.93) are extracted the coefficients appearing in Equation (2.89):

$$m_{P\,j} = \frac{\rho_j\, rps_j\, S_j}{D_{ref\,j}} \xi_j , \tag{2.94}$$

$$k_{P\,j} = \rho_j (rps_j D_{ref\,j})^2 \gamma_j . \tag{2.95}$$

2.6.3.2. Viscous losses

Viscous losses represent the irreversible transformation of mechanical energy in heat and they arise from the friction between the fluid and the circuit's walls or components. Starting from Equation (2.33) and applying the hypothesis of steady-state, fully developed incompressible flow, the terms related to the time derivative, kinetic energy difference, and volume compression work cancel out. By neglecting the difference in height and sources of mechanical energy, also the potential energy difference and the work due to viscous forces are zero. Therefore, under the mentioned hypotheses, Equation (2.33) simplifies to:

$$\int_S p\,\vec{v}\cdot\vec{n}\,dS = -\int_V \boldsymbol{\tau} : \nabla\vec{v}\,dV\ . \tag{2.96}$$

The surface integral in the LHS of Equation (2.96) is solved with the midpoint rule, considering the pressure field across the vents to be uniform. Therefore, the integral average of fluid velocity is replaced by its midpoint value, introducing a second-order error. The volume integral at the RHS, representing the viscous losses, is substituted with $-\dot{E}_v$. Thus Equation (2.96) becomes:

$$\dot{E}_v = (p_\mathrm{I} - p_\mathrm{E})\,\bar{u}\,S\ . \tag{2.97}$$

In Equation (2.97) viscous losses are expressed as a pressure drop, which is divided into two main components: losses due to fluid friction and losses due to concentrated elements in the circuit. The losses due to fluid friction are also called distributed losses, and they can be described by means of the Darcy–Weisbach equation:

$$\Delta p_\mathrm{d} = f\,\frac{L}{D_\mathrm{h}}\,\rho\,\frac{\bar{u}^2}{2}\ , \tag{2.98}$$

where f is the Darcy friction factor, which is usually a function of the Reynolds number and the surface roughness. The Reynolds number is a dimensionless quantity describing the ratio between inertial and viscous forces:

$$\mathrm{Re} = \frac{u\,D_\mathrm{h}}{\nu}\ . \tag{2.99}$$

For internal channel flow, when $\mathrm{Re} < 2000$ the regime is laminar and the friction factor does not depend on the surface roughness:

$$f = \frac{64}{\mathrm{Re}}\quad \text{, if } \mathrm{Re} < 2000\ . \tag{2.100}$$

When $\mathrm{Re} > 3000$ the turbulent regime occurs, and the surface roughness contribution becomes more important. In fact, micro-irregularities of the surface can induce flow separation, increasing the friction factor. This effect depends also on the thickness of the viscous sub-layer,

referred to the first layer of fluid in which the flow remains laminar since locally the viscous forces are more intense. When the micro-irregularities of the surface are taller than the viscous sub-layer thickness, local separation of the flow is induced; in the opposite case, when the surface irregularities are fully contained by the viscous sub-layer, the flow is hydraulically smooth. The thickness of the viscous sub-layer changes with the Reynolds number: as the Reynolds number increases, the viscous layer thickness decreases. At low Reynolds numbers, the viscous layer thickness is sufficient to contain all surface irregularities; however, as the Reynolds number increases and the viscous layer thickness decreases, the turbulent flow is exposed to the surface irregularities. For this reason, when the flow is fully turbulent, the friction factor depends only on the surface roughness. The most common correlation used to calculate the friction factor for turbulent flows is the Colebrook-White equation [134]

$$\frac{1}{\sqrt{f}} = -2.0 \log\left(\frac{Ra}{3.7\,D_\mathrm{h}} + \frac{2.51}{\mathrm{Re}\,\sqrt{f}}\right), \tag{2.101}$$

this equation is implicit since f is on both sides and is usually solved by means of iterative procedures. To avoid the implicit term in the equation, explicit approximations of Equation (2.101) were developed. These approximations are less accurate with respect to the Colebrook equation but are simpler to handle since they are explicit; an example is the Haaland [135] equation:

$$\frac{1}{\sqrt{f}} = -1.8 \log\left(\left(\frac{Ra}{3.7\,D_\mathrm{h}}\right)^{1.11} + \frac{6.9}{\mathrm{Re}}\right). \tag{2.102}$$

When the pipes are smooth, the Blasius formula gives good accuracy, even though it is only dependent on the Reynolds number:

$$f = \frac{0.316}{\mathrm{Re}^{1/4}}\,. \tag{2.103}$$

Finally, by substituting the pressure difference in Equation (2.97) with Equation (2.98), the viscous losses are formulated as:

$$\dot{E}_v = f\,\frac{L}{D_\mathrm{h}}\,\frac{1}{2}\,\rho\,\bar{u}^3\,S\,. \tag{2.104}$$

When dealing with components featuring complex geometries, it is not possible to formulate the problem as a pipe flow problem; therefore, the viscous losses across these components are expressed as concentrated losses:

$$\Delta p_\mathrm{c} = K\,\frac{1}{2}\,\rho\,\bar{u}^2\,, \tag{2.105}$$

where the parameter K is the concentrated loss coefficient which is usually extracted by means of experimental or numerical analysis. Thus, the total pressure losses are the sum of the distributed losses and concentrated losses:

$$\Delta p_{\mathrm{L}} = \Delta p_{\mathrm{d}} + \Delta p_{\mathrm{c}} = \left(f \frac{L_j}{D_{\mathrm{h}}} + K_j \right) \rho \frac{\bar{u}^2}{2} \,. \tag{2.106}$$

Therefore, for each branch j of the computational domain, the total pressure losses can be expressed as:

$$\Delta p_{\mathrm{L},j} = \left(f_j \frac{L_j}{D_{\mathrm{h},j}} + K_j \right) \frac{\rho_j \bar{u}_j \left| \bar{u}_j \right|}{2} \,, \tag{2.107}$$

where the square of the fluid velocity in Equation (2.106) is replaced with $\bar{u}_j |\bar{u}_j|$ to ensure that the pressure drop always follows the direction of fluid velocity. Since Equation (2.107) is quadratic, it needs to be linearised to be formulated as a general source/sink term:

$$\Delta p_{\mathrm{L},j} \approx \sigma_{\mathrm{L},j}^{k+1} = m_{\mathrm{L},j} \, \bar{u}_j^{k+1} + k_{\mathrm{L},j} \,, \tag{2.108}$$

where the coefficients m and k can be derived by comparing Equation (2.108) with Equation (2.107):

$$m_{\mathrm{L},j} = \left(f_j \frac{L_j}{D_{\mathrm{h},j}} + K_j \right) \frac{\rho_j \left| u_j^k \right|}{2} \,, \quad k_{\mathrm{L},j} = 0 \,. \tag{2.109}$$

2.6.3.3. Branch thermal properties

For a generic branch CV, the related thermal power is the sum of the heat transfer across the volume's boundaries and any heat generation/removal due to specific components inside the branch.

$$\dot{Q}_j = \int_S -\vec{q} \cdot \vec{n} \, dS + \int_V H \, dV \,, \tag{2.110}$$

where $\vec{q}$ is the heat transfer across the boundaries and H is a volumetric heat source/sink. The first integral of Equation (2.110) can be expressed by employing Newton's law of cooling:

$$\int_S -\vec{q} \cdot \vec{n} \, dS = U_j \, S_{\mathrm{ref},j} \left(T_\infty - \bar{T}_j \right) \,, \tag{2.111}$$

where U_j represents the global heat transfer coefficient and T_∞ is the reference temperature of the external environment. The product $U_j S_{\mathrm{ref},j}$ can be expressed as a function of the equivalent thermal resistance:

$$\frac{1}{U_j \, S_{\mathrm{ref},j}} = R_{\mathrm{eq},j} \,. \tag{2.112}$$

By excluding the presence of specific components that generate or remove heat, the second integral in Equation (2.110) is zero and the equation becomes:

$$\dot{Q}_j = \frac{\left(T_\infty - \bar{T}_j\right)}{R_{\text{eq},j}} \,. \tag{2.113}$$

As example of heat transfer process across branch volume boundaries, consider the heat transfer between the flow inside a round pipe and the external environment. In this case, the heat transfer mechanisms are the internal convection between the fluid and the pipe internal surface, the conduction through the pipe wall thickness, and finally the external convection and radiation between the pipe external surface and the environment. The equivalent resistance can be represented as a series of resistances:

$$R_{\text{eq},j} = R_{\text{int}} + R_{\text{w}} + \frac{R_{\text{ext}} R_{\text{rad}}}{R_{\text{ext}} + R_{\text{rad}}} \,, \tag{2.114}$$

the last contribution is the resistance governing the heat transfer between the external surface of the pipe and the environment, and it is expressed as a parallel of convective heat transfer and radiation, since these two mechanisms occur simultaneously. The internal convective resistance is calculated as follows:

$$R_{\text{int}} = \frac{1}{h_{\text{int}} \left(\frac{\Delta T_{\text{lm}}}{\Delta T} \right) S_{\text{int}}} \tag{2.115}$$

where h_{int} is the internal convective heat transfer coefficient and can be extrapolated from the Dittus-Boelter correlation for turbulent flow:

$$h_{\text{int}} = \frac{\lambda}{D_{\text{h}}} 0.023 \,\text{Re}^{0.8} \,\text{Pr}^{0.4} \,. \tag{2.116}$$

The conductive resistance on the pipe wall thickness is:

$$R_w = \frac{\ln\left(D_{\text{ext}} / D_{\text{h}}\right)}{2 \,\pi \,\lambda_w \,L} \,, \tag{2.117}$$

where λ_{w} is the thermal conductivity of the pipe material and D_{ext} is the tube external diameter. The external convective resistance is:

$$R_{\text{ext}} = \frac{1}{h_{\text{ext}} S_{\text{ext}}} \,, \tag{2.118}$$

where h_{ext} is the convective coefficient between the pipe external surface and the external environment. By considering the external surface of the pipe as a grey body characterised by emissivity ε_{r}, and considering also that the external surface of the pipe is much smaller than

the environment surface, the heat transfer by means of radiation is:

$$\dot{Q}_{rad} = S_{\text{ext}}\,\sigma_{\text{B}}\,\varepsilon_{\text{r}}\left(T_{\text{ext}}^4 - T_{\infty}^4\right)\,, \tag{2.119}$$

where T_{ext} is the temperature of the pipe external surface and σ_{B} is the Stefan-Boltzmann constant. Equation (2.119) can be reformulated as follows:

$$\dot{Q}_{rad} = S_{\text{ext}}\,\sigma_{\text{B}}\,\varepsilon_{\text{r}}\left(T_{\text{ext}}^2 + T_{\infty}^2\right)\left(T_{\text{ext}} + T_{\infty}\right)\left(T_{\text{ext}} - T_{\infty}\right). \tag{2.120}$$

By using the radiation heat transfer coefficient h_{rad}, Equation (2.120) becomes:

$$\begin{aligned} \dot{Q}_{rad} &= h_{\text{rad}}\,S_{\text{ext}}\,(T_{\text{ext}} - T_{\infty}) \\ h_{\text{rad}} &= \sigma_{\text{B}}\,\varepsilon_{\text{r}}\left(T_{\text{ext}}^2 + T_{\infty}^2\right)\left(T_{\text{ext}} + T_{\infty}\right). \end{aligned} \tag{2.121}$$

By using the formulation of Equation (2.121) it is easy to find the radiative resistance:

$$R_{\text{rad}} = \frac{(T_{\text{ext}} - T_{\infty})}{\dot{Q}_{\text{rad}}} = \frac{1}{h_{\text{rad}}\,S_{\text{ext}}}\,. \tag{2.122}$$

In the toolbox, any heat transfer process involving the external boundaries of a branch volume is formulated as a generalised source/sink term:

$$\dot{Q}_j^{\,k+1} \approx \sigma_{\text{e},j}^{\,k+1} = m_{\text{e},j}\,T_j^{\,k+1} + k_{\text{e},j}\,, \tag{2.123}$$

where the superscript $k+1$ refers to the value computed at the $k+1$ iteration of the solver. The coefficients m and k are derived by comparing Equation (2.123) with Equation (2.113):

$$m_{\text{e},j} = -\frac{1}{R_{\text{eq},j}}\,, \quad k_{\text{e},j} = \frac{T_{\infty}}{R_{\text{eq},j}}\,. \tag{2.124}$$

In Equation (2.124), the equivalent resistance $R_{\text{eq},j}$ is calculated by using the fields and fluid properties related to the k-th iteration of the solver, since their values at the iteration $k+1$ are unknown.

2.6.3.4. Heat exchangers

To model heat exchangers, the ε-NTU method is employed:

$$\dot{Q}_j = \int_S -\vec{q}\cdot\vec{n}\,dS = \varepsilon\,C_{\text{r,min}}\left(T_{\text{ext,in}} - \bar{T}_j\right) = -\varepsilon\,C_{\text{r,min}}\,\text{ITD}\,, \tag{2.125}$$

where ε is the effectiveness of the heat exchanger and is usually computed from correlations for the specific type of heat exchanger, and $C_{\text{r,min}}$ is the minimum heat capacity rate between

the two flows. $T_{\text{ext,in}}$ represents the inlet temperature of the external fluid, while the temperature of the branch T_j corresponds to the temperature of the upwind node, which is the inlet node of the branch. The difference $\left(T_{\text{ext,in}} - \bar{T}_j\right)$ is called Inlet Temperature Difference (ITD), and represents the maximum temperature difference between the fluids flowing inside the heat exchanger. In the toolbox the effects of heat exchangers are treated as heat source/sink terms, by employing the formulation in Equation (2.123). The m and k coefficients are derived by comparing Equation (2.123) with Equation (2.125):

$$m_{\text{e},j} = -\varepsilon\, C_{\text{r,min}}\,, \quad k_{\text{e},j} = \varepsilon\, C_{\text{r,min}}\, T_{\text{ext,in}}\,. \tag{2.126}$$

2.6.3.5. Heating and cooling elements

The uniform generation or absorption of heat within the fluid volume associated with a branch is modelled by means of a lumped heating or cooling element, as shown in Figure 2.2b:

$$\dot{Q}_j = \int_V H\, dV = \dot{Q}_{\text{g},j}\,. \tag{2.127}$$

To express Equation (2.127) using the generic source formulation given in Equation (2.123), the coefficients m and k are defined as:

$$m_{\text{e},j} = 0\,, \quad k_{\text{e},j} = \dot{Q}_{\text{g},j}\,. \tag{2.128}$$

2.6.4. Solution algorithm

In the fluid flow solver, the primary variables are velocity, pressure, and temperature. To determine the values of these variables at each node or branch, three governing equations are solved: the macroscopic mass balance, the macroscopic mechanical energy balance, and the macroscopic thermal energy balance.

$$\begin{cases} \boldsymbol{A}\boldsymbol{M}\,\boldsymbol{u}^{n+1} = -\boldsymbol{d} \\ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^{n+1} = \boldsymbol{Y}\,\boldsymbol{u}^{n+1} + \boldsymbol{k} \\ \boldsymbol{W}\,\boldsymbol{T}^{n+1} = \boldsymbol{s} \end{cases}. \tag{2.129}$$

Pressure and velocity are both present in the mechanical energy balance equation; thus, there are two variables in one equation. Since the mass balance equation acts only as a constraint, there is no other equation that describes the evolution of velocity or pressure. Therefore, it is necessary to decouple these two variables. This is achieved using an algorithm based on the SIMPLE algorithm [129], which is an iterative procedure developed starting from the pressure correction method [136]. The main steps of the SIMPLE algorithm are:

i. Pressure and velocity are decomposed into guessed φ^* and correction φ' fields; the

algebraic sum of the guessed value and the correction value results in the true value:

$$\boldsymbol{P} = \boldsymbol{P}^* + \boldsymbol{P}' \,, \tag{2.130}$$

$$\boldsymbol{u} = \boldsymbol{u}^* + \boldsymbol{u}' \,. \tag{2.131}$$

ii. The guessed pressure field $\boldsymbol{P}^*$ is initialised with a guess value.

iii. The guessed pressure field $\boldsymbol{P}^*$ is used in the mechanical energy equation to derive the guessed velocity field $\boldsymbol{u}^*$.

iv. The continuity equation is solved with the guessed velocity field $\boldsymbol{u}^*$. The continuity equation is likely not satisfied by the guessed velocity; in this case, the residual of the continuity equation is used to derive a pressure correction field $\boldsymbol{P}'$.

v. Use the pressure correction field $\boldsymbol{P}'$ to derive the velocity correction field $\boldsymbol{u}'$.

vi. Use the correction fields to correct velocity and pressure and repeat the procedure from step 3 until the continuity equation is respected within a specified tolerance.

2.6.4.1. Derivation of the pressure correction equation

To develop the SIMPLE-based algorithm to decouple pressure and velocity, it is necessary to derive the pressure correction equation.
By using the guessed pressure field in Equation (2.62), the guessed velocity field is obtained:

$$\boldsymbol{u}^* = \boldsymbol{Y}^{-1} \left(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{P}^* - \boldsymbol{k} \right) \,. \tag{2.132}$$

By decomposing the pressure and velocity fields as in Equations (2.131) and (2.130), it is possible to derive a relation between the correction fields from Equation (2.62):

$$\boldsymbol{A}^{\mathrm{T}} \left(\boldsymbol{P}^* + \boldsymbol{P}' \right) = \boldsymbol{Y} \left(\boldsymbol{u}^* + \boldsymbol{u}' \right) + \boldsymbol{k} \,. \tag{2.133}$$

The next step is to pre-multiply all terms by $\boldsymbol{Y}^{-1}$ and substitute the guessed velocity field $\boldsymbol{u}^*$ with the relation in Equation (2.132):

$$\boldsymbol{Y}^{-1} \left[\boldsymbol{A}^{\mathrm{T}} \left(\boldsymbol{P}^* + \boldsymbol{P}' \right) \right] = \boldsymbol{Y}^{-1} \left[\boldsymbol{Y} \left(\boldsymbol{Y}^{-1} \left(\boldsymbol{A}^T \boldsymbol{P}^* - \boldsymbol{k} \right) + \boldsymbol{u}' \right) + \boldsymbol{k} \right] \,, \tag{2.134}$$

after performing some simplifications, an equation for the velocity correction field depending on the pressure correction field remains:

$$\boldsymbol{u}' = \boldsymbol{Y}^{-1} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{P}' \,. \tag{2.135}$$

Similar to the mechanical energy balance, the continuity equation (Equation (2.42)) can also be rearranged by decomposing the velocity field according to Equation (2.131):

$$\boldsymbol{A}\boldsymbol{M}\left(\boldsymbol{u}^{*}+\boldsymbol{u}^{\prime}\right)=-\boldsymbol{d}\ . \tag{2.136}$$

By substituting Equation (2.135) for the velocity correction field in Equation (2.136), the pressure correction equation is obtained:

$$\boldsymbol{\Lambda}\,\boldsymbol{P}^{\prime}=\boldsymbol{b}\ , \tag{2.137}$$

where $\boldsymbol{\Lambda}$ is an $n_{\mathrm{n}}\times n_{\mathrm{n}}$ matrix calculated as:

$$\boldsymbol{\Lambda}=\boldsymbol{A}\,\boldsymbol{M}\,\boldsymbol{Y}^{-1}\boldsymbol{A}^{\mathrm{T}}\ , \tag{2.138}$$

and $\boldsymbol{b}$ is the known vector with elements:

$$\boldsymbol{b}=-\boldsymbol{A}\,\boldsymbol{M}\,\boldsymbol{u}^{*}-\boldsymbol{d}\ . \tag{2.139}$$

2.6.4.2. Boundary conditions for the pressure field

At the boundary nodes, defined as nodes connected to a single branch, the pressure is set to a fixed or time-varying value. To enforce pressure boundary conditions, the prescribed value is assigned to the guessed pressure field, while the corresponding pressure correction value is set to zero. Therefore, at the boundary nodes, Equation (2.130) becomes:

$$\boldsymbol{P}_{\mathrm{bou}}=\boldsymbol{P}^{*}_{\mathrm{bou}}+\boldsymbol{P}^{\prime}_{\mathrm{bou}}\qquad\Rightarrow\qquad\boldsymbol{P}^{\prime}_{\mathrm{bou}}=\boldsymbol{0}\ . \tag{2.140}$$

Since every boundary node is connected to a single branch, according to Equation (2.4), the corresponding rows in the incidence matrix contain only one non-zero element. Thus, boundary nodes are identified using the $n_{\mathrm{n}}\times n_{\mathrm{n}}$ diagonal matrix $\boldsymbol{I}_{\mathrm{bou}}$:

$$I_{\mathrm{bou},ii}=1\qquad\text{if}\qquad\sum_{j}^{n_{\mathrm{b}}}|A_{ij}|=1\ . \tag{2.141}$$

To apply pressure boundary conditions, the pressure correction equation (Equation (2.137)) is modified to satisfy the relations in Equation (2.140) at the boundary nodes:

$$\boldsymbol{\Lambda}_{\mathrm{bc}}\boldsymbol{P}^{\prime}=\boldsymbol{b}_{\mathrm{bc}}\ , \tag{2.142}$$

where

$$\boldsymbol{\Lambda}_{\mathrm{bc}}=\boldsymbol{I}_{\mathrm{int}}\boldsymbol{\Lambda}+\boldsymbol{I}_{\mathrm{bou}}\ , \tag{2.143}$$

$$\boldsymbol{b}_{\mathrm{bc}} = \boldsymbol{I}_{\mathrm{int}}\,\boldsymbol{b}\,. \tag{2.144}$$

In Equation (2.143), $\boldsymbol{I}_{\mathrm{int}}$ is the matrix identifying the internal nodes, obtained by applying the logical not operation to each element of $\boldsymbol{I}_{\mathrm{bou}}$. In the modified coefficient matrix ($\boldsymbol{\Lambda}_{\mathrm{bc}}$), every element in a row corresponding to a boundary node is first set to zero; subsequently, by adding $\boldsymbol{I}_{\mathrm{bou}}$, the diagonal element of that row is set to 1. The elements of $\boldsymbol{b}$ related to boundary nodes are set to zero by multiplying $\boldsymbol{b}$ by the matrix identifying internal nodes.

2.6.4.3. Boundary conditions for the temperature field

The boundary conditions for the temperature field are applied at the inlet and outlet boundary nodes: at the inlet, a fixed temperature value is imposed, while at the outlet, a zero-gradient condition is set. The inlet and outlet boundary nodes are identified based on the sign of the fluid velocity in their respective branches. These conditions modify Equation (2.77) at the boundary nodes to obtain:

$$\begin{cases} \bar{T}_i = T_{i,\mathrm{bc}} & \text{if node } i \text{ is an inlet point} \\ \bar{T}_i = \sum_j^{n_\mathrm{b}} |A_{ij}|\,T_j & \text{if node } i \text{ is an outlet point} \end{cases}. \tag{2.145}$$

Equation (2.145) is enforced in the system of equations by modifying Equation (2.77):

$$\boldsymbol{W}_{\mathrm{bc}}\,\boldsymbol{T} = \boldsymbol{s}_{\mathrm{bc}}\,, \tag{2.146}$$

where

$$\boldsymbol{W}_{\mathrm{bc}} = \boldsymbol{I}_{\mathrm{int}}\,\boldsymbol{W} + \boldsymbol{I}_{\mathrm{bou}}\,, \tag{2.147}$$

and

$$s_{\mathrm{bc},i} = \begin{cases} \dfrac{1}{2}\sum_j^{n_\mathrm{b}} \left[\left|A_{ij}\right| \left(T_{i,\mathrm{bc}} + \bar{T}_j\right) + \dfrac{\bar{u}_j}{\left|\bar{u}_j\right|} \left(T_{i,\mathrm{bc}} - \bar{T}_j\right) \right] & \text{for boundary nodes} \\ s_i & \text{for internal nodes} \end{cases}. \tag{2.148}$$

In Equation (2.147), the coefficient matrix is modified following the same reasoning as Equation (2.143), while the known terms vector $s_{\mathrm{bc},i}$ sets the temperature boundary value if the velocity is positive (inlet node), otherwise it sets the value of the neighbour branch, applying the zero-gradient condition.

2.6.4.4. The steady-state flow solver

The steady-state flow solver solves the presented governing equations by neglecting the time-dependent terms. The steps of the solver are:

i. Setting of the boundary conditions and initial values for each field, including the

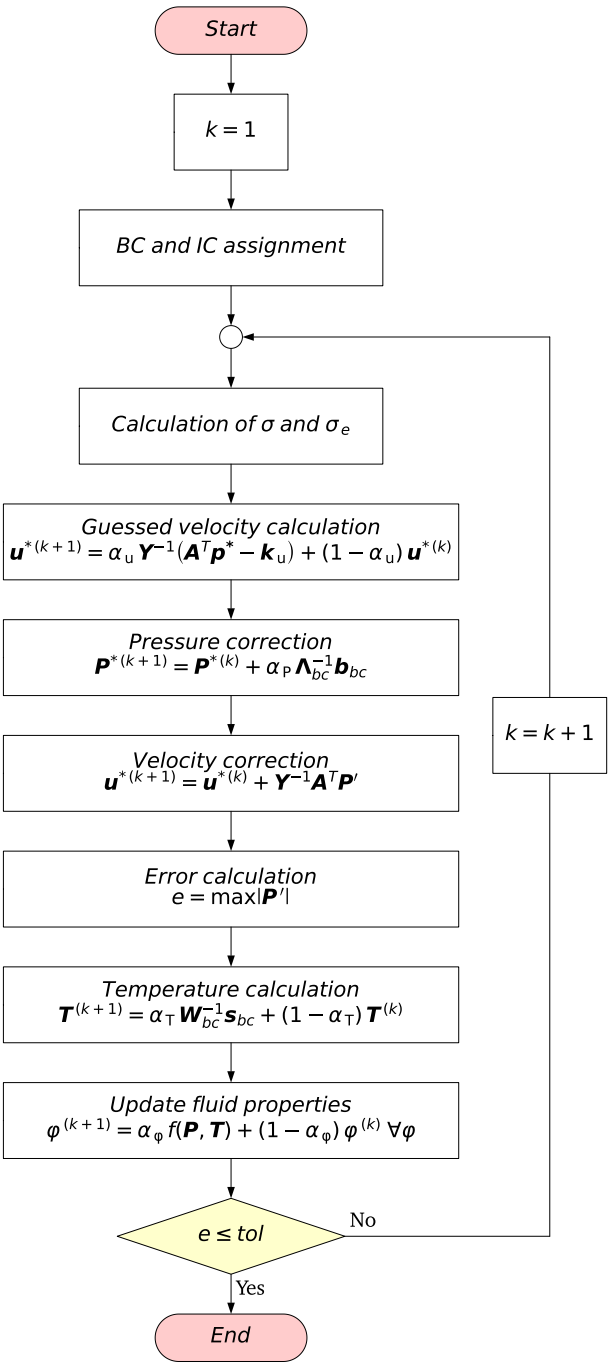


Figure 2.5: Flow chart of the steady-state flow solver algorithm, where k represents the iteration number.

physical properties:

$$\varphi_r = \varphi_{\text{init}} , \quad \varphi_r = \varphi_{\text{bc},r} \quad \forall\, \varphi, r . \tag{2.149}$$

Since the solver is steady-state, the initial values are arbitrarily decided; in fact, the initial values do not affect the final solution, but only how many iterations the solver takes.

ii. The solver calculates the coefficients m_{r} and k_{r} for each assigned source/sink according to the model used. These terms are calculated for all nodes and branches in which a source/sink is assigned.

iii. Solution of the mechanical energy balance to derive a guessed velocity field from the guessed pressure field:

$$\boldsymbol{u}^*_{\text{new}} = \boldsymbol{Y}^{-1} \left(\boldsymbol{A}^{\text{T}} \boldsymbol{P}^* - \boldsymbol{k}_{\text{u}} \right) . \tag{2.150}$$

To damp the oscillations of the velocity values across the iterations, an under-relaxation factor α_{u} is applied:

$$\boldsymbol{u}^{*(k+1)} = \alpha_{\text{u}}\, \boldsymbol{u}^*_{new} + (1-\alpha_{\text{u}})\, \boldsymbol{u}^{*(k)} . \tag{2.151}$$

iv. Pressure correction field is obtained by solving Equation (2.142), and the pressure is corrected and relaxed:

$$\boldsymbol{P}'_{new} = \boldsymbol{\Lambda}^{-1}_{\text{bc}}\, \boldsymbol{b}_{\text{bc}} , \quad \boldsymbol{P}^{*(k+1)} = \boldsymbol{P}^{*(k)} + \alpha_{\text{P}}\, \boldsymbol{P}'_{new} . \tag{2.152}$$

The new corrected pressure field is the guessed pressure for the next iteration.

v. By solving Equation (2.135), the velocity correction field is obtained. Then, the velocity is corrected:

$$\boldsymbol{u}'_{new} = \boldsymbol{Y}^{-1} \boldsymbol{A}^{\text{T}} \boldsymbol{P}' , \quad \boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{*(k+1)} + \boldsymbol{u}'_{new} . \tag{2.153}$$

vi. The error of the pressure-velocity decoupling algorithm is evaluated by taking the maximum value of the pressure correction field in absolute value:

$$e = \max \left| \boldsymbol{P}' \right| . \tag{2.154}$$

vii. Solution of Equation (2.146) to find the new temperature field:

$$\boldsymbol{T}_{\text{new}} = \boldsymbol{W}^{-1}_{\text{bc}}\, \boldsymbol{s}_{\text{bc}} , \quad \boldsymbol{T}^{(k+1)} = \alpha_{\text{T}}\, \boldsymbol{T}_{\text{new}} + (1-\alpha_{\text{T}})\, \boldsymbol{T}^{(k)} . \tag{2.155}$$

viii. Updating of the fluid property fields according to the state equations or by interpolating tabulated data as a function of pressure and temperature. The property fields are

relaxed in each update:

$$\varphi_{\text{new},i} = f\left(P_i,\, T_i\right) , \quad \boldsymbol{\varphi}^{(k+1)} = \alpha_{\varphi}\, \boldsymbol{\varphi}_{new} + (1-\alpha_{\varphi})\, \boldsymbol{\varphi}^{(k)} \quad \forall\, \varphi,\, i\, . \tag{2.156}$$

ix. If the error e is lower than the tolerance set, the solver stops and returns all the node and branch fields. Otherwise, the procedure is repeated from step 2.

Figure 2.5 shows the flow chart describing the steady-state flow solver, with all the steps already mentioned.

2.6.4.5. The unsteady solver

In the unsteady solver, the governing equations are solved including all terms. At each time-step, the solution of that time-step is attained using the same procedure employed in the steady-state solver. The steps of the unsteady solver are:

i. Setting initial values for each field. In this case, the initial values can affect the final solution.

ii. Setting boundary conditions, which can also be time-varying functions. To handle time-varying conditions, the boundary conditions are updated at the start of each time-step.

iii. Solving time-step n with the same iterative algorithm used in the steady-state solver; the only difference is that in this case also the time-dependent terms are considered in the governing equations.

iv. Temporal progression:

$$t^{n+1} = t^{n} + \Delta t\, . \tag{2.157}$$

After updating the time-step, the procedure re-starts from step 2 to find the solution of the new time-step. This loop is repeated until all the time-steps are solved.

The algorithm used in the unsteady solver is represented as a flow chart in Figure 2.6, where the inner loop employed to solve the single time-step is within a rectangular box.

2.7. Implementation of a thermal network solver

In many thermal-fluid system analyses, the interest focuses mainly on thermal performance, and solving for the velocity and pressure fields is not required. In these applications, a thermal network solver offers a simpler and more flexible approach to model thermal systems, such as battery pack cooling systems [137], electric motors [138], and high-frequency power transformers [139]. The Lumped Parameter Thermal Network (LPTN) approach is based on

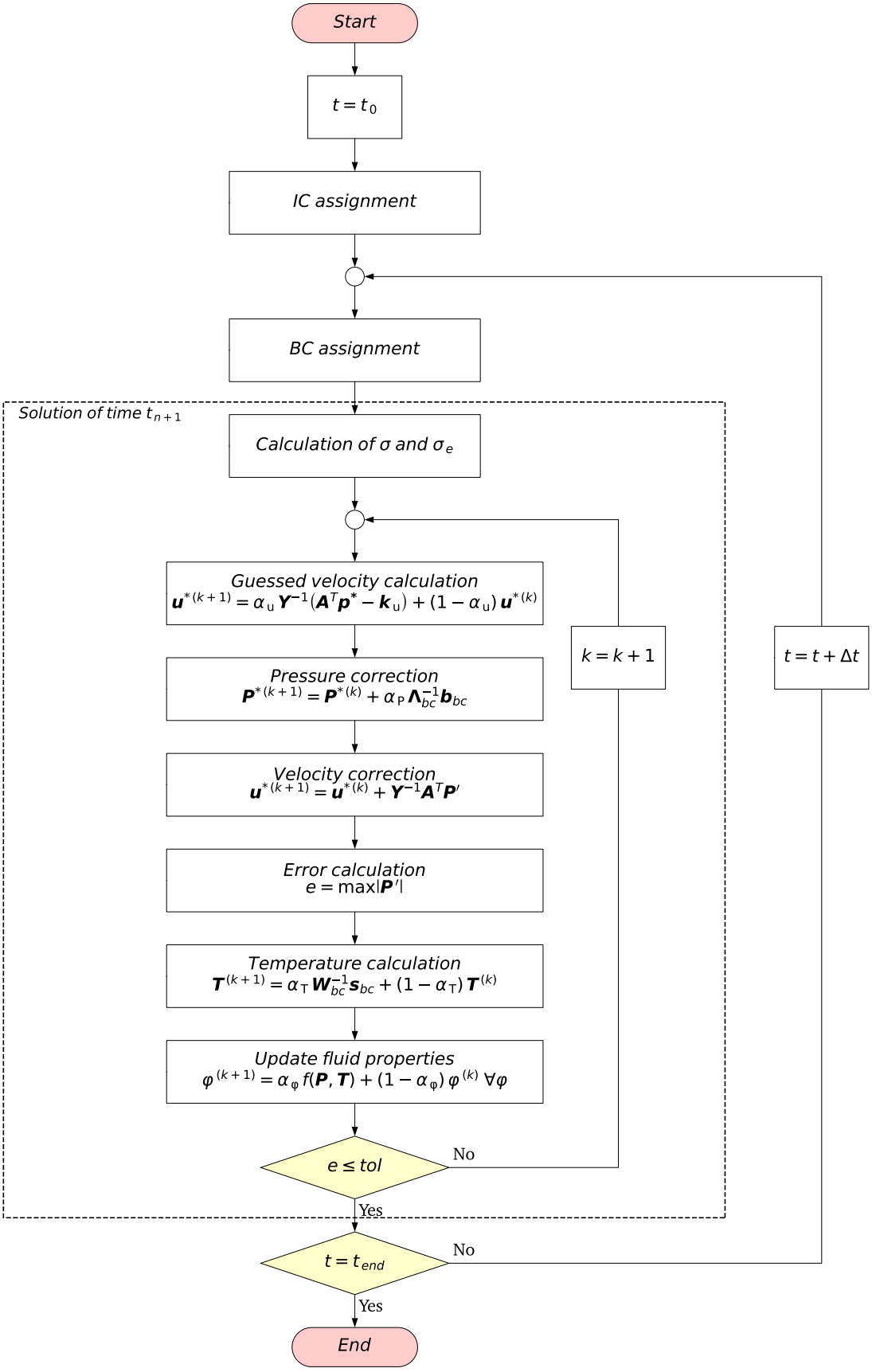


Figure 2.6: Flow chart of the unsteady flow solver algorithm.

the electro-thermal analogy:

$$\Delta V = I R_{el} \Rightarrow \Delta T = \dot{Q} R_T \,, \tag{2.158}$$

where the relation between potential difference, current intensity and electric resistance described by Ohm's law is used to relate temperature difference, heat and thermal resistance. In LPTNs, the physical domain is divided into a set of regions, and a node is placed in the center of each region. The temperature of the node represents the mean temperature of the lumped region. Given this, the subdivision should be made by placing the nodes in zones where low temperature gradients are expected, since the LPTN is not able to capture the gradients inside the lumped regions. The level of accuracy of these models depends on the level of discretisation: the more the domain is discretised, the more temperature gradients are captured. The nodes are connected by means of thermal resistances representing the main path followed by the heat flux; the type of thermal resistance depends on the heat transfer mechanism described by the resistance itself. The graph-based domain representation discussed in Section 2.3, the generalised source term formulation described in Section 2.5, and the units of measurement treatment defined in Section 2.4 are employed to build the thermal network solver.

2.7.1. Governing equations

The only equation solved by the thermal network solver is the heat balance equation at each node i:

$$C_i \frac{dT_i}{dt} + \sum_r \frac{T_i - T_r}{R_{ir}} = \dot{Q}_i \,, \tag{2.159}$$

where the index r indicates all the nodes connected with the node i by the thermal resistance R_{ir}, $\dot{Q}_i$ is the possible heat source defined in the node i and C_i is the thermal capacity of the mass lumped in the node i. In the present toolbox thermal resistances are defined as heat sources in the branches; in fact, the heat flowing in the generic branch j is calculated on the basis of the thermal resistance described by the branch itself:

$$\dot{Q}_j = \frac{\Delta T_j}{R_j} = G_j \Delta T_j \,, \tag{2.160}$$

where G_j is the thermal conductance and it is defined as the inverse of the thermal resistance. To write the thermal resistance as a generalised source term like in Equation (2.27) it is necessary to define the coefficients m and k. These coefficients can be defined by comparing Equation (2.160) with Equation (2.27):

$$m_j = G_j \,, \quad k_j = 0 \,, \tag{2.161}$$

therefore the source term is calculated as:

$$\sigma_j = \dot{Q}_j = m_j \, \Delta T_j \, . \tag{2.162}$$

The heat generated within a node is also modelled as a generalised source term; for example, to model a constant heat source in the node i the source term is:

$$\sigma_i = \dot{Q}_i \, , \tag{2.163}$$

therefore the coefficients for the constant heat source are:

$$m_i = 0 \, , \quad k_i = \dot{Q}_i \, . \tag{2.164}$$

At this point it is convenient to substitute the sum over the r nodes in Equation (2.159) with the sum of the heat fluxes from all the branches of the network:

$$C_i \frac{dT_i}{dt} + \sum_j^{n_\mathrm{b}} A_{\mathrm{ud},ij} \, m_j \, \Delta T_j = k_i \, , \tag{2.165}$$

where $A_{\mathrm{ud},ij}$ indicates the elements of the unidirectional incidence matrix $\boldsymbol{A}_{\mathrm{ud}}$ defined in Equation (2.7). Since the unidirectional incidence matrix $\boldsymbol{A}_{\mathrm{ud}}$ is used in the governing equation, the following sign convention is introduced: the heat flux is positive if entering or generating in the node i, and is negative if exiting. The temperature difference between the nodes connected by branches can be defined by using the transpose of the incidence matrix $\boldsymbol{A}$:

$$\Delta T_j = \sum_l^{n_\mathrm{n}} A_{jl}^T \, T_l \, , \tag{2.166}$$

therefore Equation (2.165) becomes:

$$C_i \frac{dT_i}{dt} + \sum_j^{n_\mathrm{b}} A_{\mathrm{ud},ij} \, m_j \sum_l^{n_\mathrm{n}} A_{jl}^T \, T_l = k_i \, . \tag{2.167}$$

The unsteady term is discretised by using the Implicit Euler method (see Appendix A.3):

$$\frac{C_i}{\Delta t} \, T_i^{\,n+1} + \sum_j^{n_\mathrm{b}} A_{\mathrm{ud},ij} \, m_j \sum_l^{n_\mathrm{n}} A_{jl}^T \, T_l^{\,n+1} = k_i + \frac{C_i}{\Delta t} \, T_i^{\,n} \, , \tag{2.168}$$

where

$$\Delta t = t^{\,n+1} - t^{\,n} \, . \tag{2.169}$$

Since Equation (2.168) is solved in all the nodes of the domain, the following system of equations is obtained:

$$\boldsymbol{L}\,\boldsymbol{T} = \boldsymbol{r}\ , \tag{2.170}$$

where $\boldsymbol{L}$ is the $n_\mathrm{n} \times n_\mathrm{n}$ system matrix calculated as:

$$\boldsymbol{L} = \boldsymbol{L}_\mathrm{u} + \boldsymbol{A}_\mathrm{ud}\,\boldsymbol{M}_\mathrm{s}\,\boldsymbol{A}^T\ . \tag{2.171}$$

In Equation (2.171) appear the $n_\mathrm{n} \times n_\mathrm{n}$ diagonal matrix $\boldsymbol{L}_\mathrm{u}$ and the $n_\mathrm{b} \times n_\mathrm{b}$ diagonal matrix $\boldsymbol{M}_\mathrm{s}$; their elements are:

$$L_{\mathrm{u},ii} = \frac{C_i}{\Delta t}\ , \tag{2.172}$$

$$M_{\mathrm{s},jj} = m_j\ . \tag{2.173}$$

Finally, the known term vector elements are:

$$r_i = k_i + \frac{C_i}{\Delta t}\,T_i^n\ . \tag{2.174}$$

2.7.2. Modelling schemes for thermal resistances

Thermal resistances connect the temperature nodes, and their values depend on the heat transfer mode: conduction, convection, radiation, or a combination of these. As previously mentioned, in the toolbox, thermal resistances are treated as source terms on branches, where the linear coefficient m_j is equal to the conductance G_j and the known coefficient k_j is zero, as shown in Equation (2.161). Therefore, the heat transfer through a thermal resistance is the linear coefficient (equal to the conductance in this case) multiplied by the temperature difference of the connected nodes, as in Equation (2.162). To define conductive resistances, the conduction shape factor $S_{\mathrm{f},j}$ is used:

$$R_j = \frac{1}{S_{\mathrm{f},j}\,\lambda} \quad \Rightarrow \quad G_j = m_j = S_{\mathrm{f},j}\,\lambda\ , \tag{2.175}$$

where the shape factor indicates the influence of the geometry on the heat transfer and is defined as the heat transfer area divided by the heat transfer length. For instance, in the case of conduction through a plane wall, the shape factor is the surface area of the wall divided by the wall thickness:

$$S_{\mathrm{f},j} = \frac{A_j}{s_j}\ . \tag{2.176}$$

When the heat transfer mechanism is convection, radiation, or both, the heat transfer coefficient is used, which is derived from user-defined correlations or based on the branch mean temperature and temperature difference. For instance, when both convection and

radiation are present, the thermal resistance is:

$$R_j = \frac{1}{(U_c + U_r)A_{ref}} \quad \Rightarrow \quad G_j = m_j = (U_c + U_r)A_{ref}\,. \tag{2.177}$$

In Equation (2.177), U_c and U_r represent the convective and radiative heat transfer coefficients, respectively.

In many forced convection problems, it is necessary to represent the temperature change of the fluid during the flow. For instance, in heat exchangers, the inlet temperature differs from the temperature at the middle of the system, and the latter differs from the outlet temperature. To simulate the temperature changes of the fluid along its path, advective resistances are used. These resistances connect only fluid nodes and arise from the internal energy balance of a fluid flowing in a simple duct, where the heat transfer is equal to the heat capacity rate multiplied by the temperature difference between the outlet and the inlet of the duct. Therefore, when the fluid path is divided into multiple nodes connected by advective resistances, it is as if every resistance represents a duct where the inlet and outlet are the nodes connected by the branch:

$$R_{advj} = \frac{\Delta T_j}{\dot{Q}_j} = \frac{T_{i+1} - T_i}{\dot{m}_j\, c_{p,j}(T_{i+1} - T_i)} = \frac{1}{\dot{m}_j\, c_{p,j}}\,. \tag{2.178}$$

Since the heat transfer direction is known a priori in forced convection problems, advective resistances are defined only in unidirectional branches (Equation (2.7)), which allow the heat flux only in one direction. In cases where the fluid absorbs heat during the flow, the heat transfer direction in the fluid is opposite to the flow direction. This is because, going towards the outlet, the fluid temperature increases, and heat transfer goes from the node with higher temperature to the node with lower temperature.

2.7.3. Boundary conditions

In the present solver, the only imposed boundary condition is the temperature value at specified nodes. In this context, boundary conditions are not necessarily applied to topological boundary nodes as in the flow solver, but can also be imposed on topological internal nodes. This is because, in many applications of the solver, the temperature is known at nodes representing internal regions rather than at physical boundaries. When it is necessary to impose a heat flux, no dedicated boundary condition is required, as this is managed through the source term. In fact, to define a heat transfer value at a boundary node, the user simply defines the heat source at that node, as in Equation (2.163). For a zero-gradient condition, it is sufficient to define a zero heat transfer source.

To apply boundary conditions to the system of equations, $n_n \times n_n$ diagonal matrices are used,

similar to Section 2.6.4.2:

$$\begin{cases} I_{\text{bou},ii} = 1\,,\; I_{\text{int},ii} = 0 & \forall\, i \in \mathbb{N}_\text{b} \\ I_{\text{bou},ii} = 0\,,\; I_{\text{int},ii} = 1 & \forall\, i \notin \mathbb{N}_\text{b} \end{cases}. \tag{2.179}$$

In Equation (2.179) $\mathbb{N}_\text{b}$ is the set of nodes where the temperature is fixed. To select the nodes where the boundary conditions are applied, the rows of the system matrix corresponding to the nodes with a fixed temperature condition are set to zero; subsequently, a value of one is added to the diagonal elements corresponding to those nodes:

$$\boldsymbol{L}_\text{bc} = \boldsymbol{I}_\text{int}\,\boldsymbol{L} + \boldsymbol{I}_\text{bou}\,. \tag{2.180}$$

The known terms vector is modified to set the fixed temperature values:

$$\boldsymbol{r}_\text{bc} = \boldsymbol{I}_\text{int}\,\boldsymbol{r} + \boldsymbol{I}_\text{bou}\,\boldsymbol{T}_\text{bc}\,, \tag{2.181}$$

where $\boldsymbol{T}_\text{bc}$ is the vector collecting the imposed temperatures.

2.7.4. Solution algorithm

The governing equation is solved iteratively. At each time-step, internal iterations are performed to determine the temperature field, updating the values of temperature dependent thermal resistances and heat sources. Convergence is reached when the mean square error between the temperature field of the current iteration and the temperature field of the previous iteration is below a specified tolerance. When the thermal resistances and heat sources do not depend on temperature, a single iteration is sufficient to solve the time-step, as the solution is analytical. Figure 2.7 presents the flow chart of the unsteady solver, where time advancement is performed using the Implicit Euler method within an external loop. For steady-state analyses, the unsteady terms in the governing equation are neglected, and only the internal loop, indicated by a dashed box in Figure 2.7, is employed.

2.8. Concluding remarks

This chapter detailed the development of a computational toolbox designed for the creation and analysis of reduced-order models of thermo-fluid systems.

The solvers of the toolbox are developed on three key elements: graph theory provides the topological framework for the computational domain, while a robust unit management system ensures dimensional consistency across all operations; these are complemented by a generalised semi-implicit formulation for source and sink terms, which enables the modelling of both linear and non-linear physical phenomena.

These fundamental blocks enabled the creation of solvers for fluid flow and thermal net-

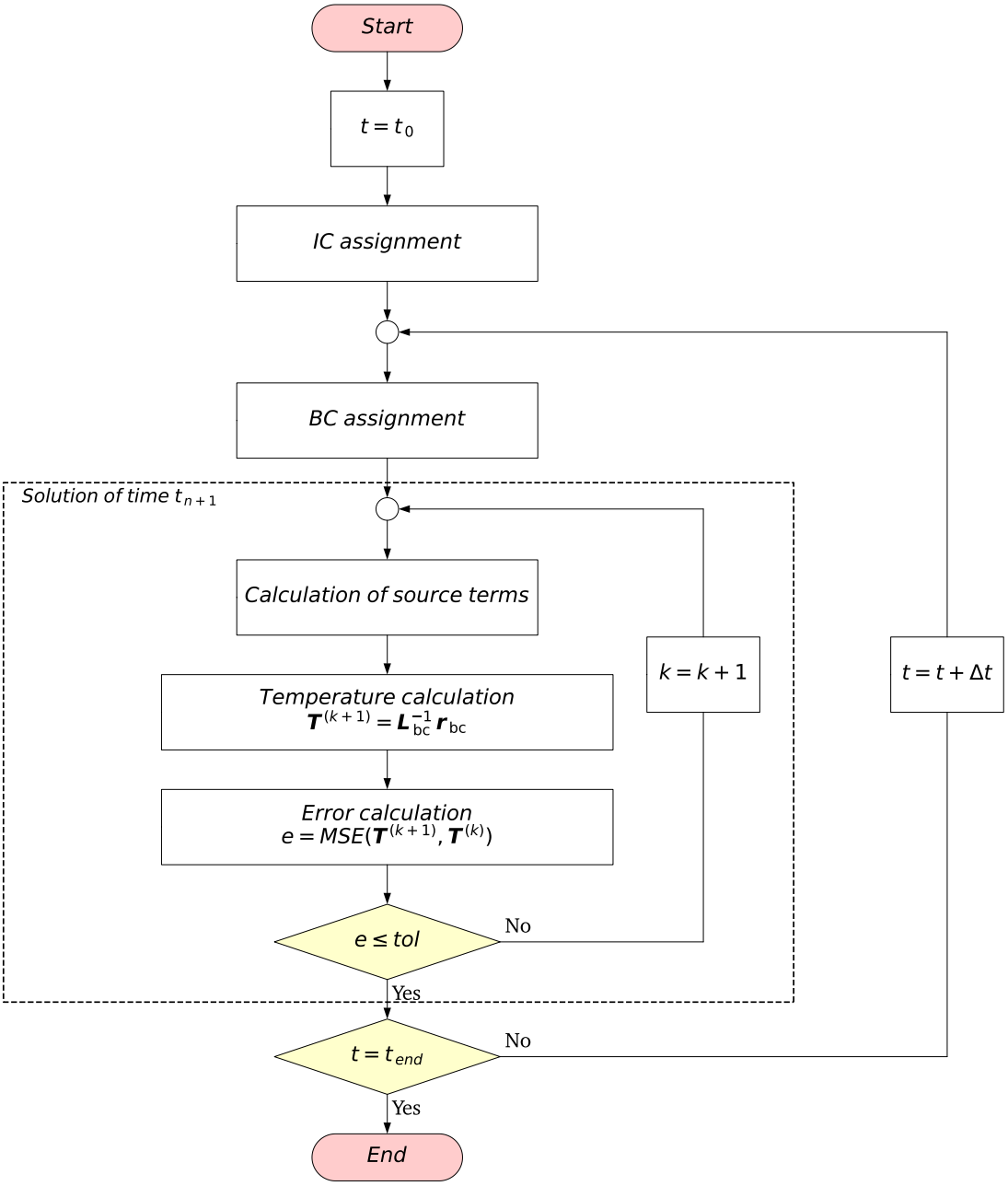


Figure 2.7: Flow chart of the solution algorithm for transient Lumped Parameters Thermal Networks (LPTNs).

works. The fluid flow solver adopts a one-dimensional Finite Volume approach, where typical pipe network components are modelled via the generalised source and sink formulation. To address the coupling between pressure and velocity fields, both steady-state and transient solvers utilise the SIMPLE algorithm.

The thermal network solver exploits a Lumped Parameter Thermal Network (LPTN) approach. Within this framework, thermal resistances representing conduction, convection, radiation, and advection are modelled as source terms located on the branches. Since material properties, thermal resistances and heat sources may vary with temperature, an iterative procedure is employed to manage these non-linearities.

Future research dedicated to this toolbox will focus on the development of multi-physics solvers, the integration of a dynamic time-stepping scheme and the compilation of specific library sections to reduce the computational burden associated with large-scale models.

CHAPTER 3

Multiscale modeling of an integrated battery thermal management system

3.1. Introduction

In the current landscape of electric mobility, the demand for extended driving range and faster charging times drives the development of battery packs with increasingly higher energy capacities and power electronic converters capable of handling higher power levels. Traditionally, the battery pack and power electronics (such as the traction inverter, on-board charger, and DC/DC converter) are designed as separate units, each with its own housing and thermal management system. However, this segregated approach results in significant volume occupation, increased weight, and complex electrical connections, which limits the overall efficiency of the vehicle. To address these limitations, recent trends in automotive engineering are moving towards the physical and functional integration of these components. This chapter focuses on the numerical modelling of a Battery Integrated Converter (BIC), a highly integrated system where the energy storage system and the power electronic converter are merged into a single compact unit. The core element of this design is a shared liquid cooling plate that serves a dual purpose: it acts as a structural support and a common heat sink for both subsystems. The work discussed in this chapter is part of the SCAPE project, supported by the European Union's Horizon Europe programme under grant agreement No. 101056781. The project aims to develop the next generation of power electronics for electric vehicles, focusing on modularity, high efficiency, and high power density. The specific configuration analysed in this work features a "sandwich" layout: the battery modules are mounted on the top surface of a cooling plate, while the power electronic components, comprising High Voltage (HV) and Low Voltage (LV) Switching Cells (SCs) and magnetic components, are attached to the bottom surface. This architecture maximises volumetric

NOMENCLATURE

Latin letters

A	Surface area [m^2]
C	Electrical capacitance [F]
c_p	Specific heat [$J\,kg^{-1}\,K^{-1}$]
dV_{oc}/dT	Entropic coefficient [$V\,K^{-1}$]
e	Error
h	Specific enthalpy [$J\,Kg^{-1}$]
i	Current intensity [A]
K	Concentrated loss coefficient
K_I	Controller integral constant
K_P	Controller proportional constant
L	Controller output
$\dot{m}$	Mass flow rate [$kg\,s^{-1}$]
p	Pressure [Pa]
$\mathscr{Q}$	Battery capacity [C]
$\dot{Q}$	Heat [W]
q	Heat flux [$W\,m^{-2}$]
R	Electrical resistance [Ω]
R_{th}	Thermal resistance [$K\,W^{-1}$]
Re	Reynolds number
S	Shape factor [m]
s	Thickness [m]
SOC	State of charge
t	Time-step index
T	Temperature [K]
U	Convective heat transfer coefficient [$W\,m^{-2}\,K^{-1}$]
$\boldsymbol{u}$	Velocity [$m\,s^{-1}$]
V_{oc}	Open-circuit voltage [V]
v	Voltage [V]

Greek letters

λ	Thermal conductivity [$W\,m^{-1}\,K^{-1}$]
ν	Laminar kinematic viscosity [$m^2\,s^{-1}$]
ν_t	Turbulent kinematic viscosity [$m^2\,s^{-1}$]
ρ	Density [$kg\,m^{-3}$]
σ	Momentum source [$m\,s^{-2}$]
τ	Time constant [s]
$\boldsymbol{\tau}$	Viscous stress tensor [Pa]

Subscripts

adv	Advective
B	Battery property
cond	Conductive
conv	Convective
fp	flat-plate
gen	Generated
J	Joule
nom	Nominal
p	Polarisation
r	Reaction

power density but introduces complex thermal coupling challenges. The problem is addressed with a multi-scale approach capable of balancing accuracy with computational cost. Given the high complexity of the assembly, which consists of a large number of distinct components, a full 3D Computational Fluid Dynamics (CFD) simulation of the complete system would incur an excessive computational burden. Consequently, the proposed solution addresses the problem on distinct scales. At the chemical reaction scale, an electrochemical model is employed to predict the heat generation within the battery cells, capturing the thermal effects of the internal chemical reactions, including entropic heat. At the cooling system scale, a CFD simulation is employed on a periodic block of the cooling channel to characterise the detailed

thermo-hydraulic behaviour of the fluid and provide accurate local heat transfer coefficients. Finally, at the full system scale, a Lumped Parameter Thermal Network (LPTN) integrates the inputs from the previous scales to model the heat transfer across the complete assembly. This approach allows for efficient unsteady simulations of the global system to verify if the cooling system is capable of maintaining all components within their safe operating limits. This chapter describes the development of this electro-thermal model using the custom library described in Chapter 2 and the integration of experimental data for battery heat generation. The goal is to verify that the proposed integrated cooling strategy maintains all sub-components within their safe operating limits under realistic dynamic loads.

3.2. Literature Survey

In modern Battery Electric Vehicle (BEV) design, the integration of power electronics directly into the battery pack represents a shift towards higher system density. This architecture physically merges the energy storage system with the converters, effectively removing the need for connections between separate units. The primary advantage of this physical integration is the reduction of resistive losses and weight, which contributes to overall vehicle efficiency and compactness [140]. Beyond physical compactness, integration offers significant functional advantages through component consolidation. As detailed by Balachandran [141], conventional battery packs are often limited by the "weakest cell" due to fixed series connections. By implementing Battery-Integrated Modular Multilevel Converters (BI-MMC), the power electronics can dynamically reconfigure the battery connections. This granular control not only extends the battery pack's lifetime by balancing individual cell loads but also enables integrated charging. Similarly, Reddy and Das [142] demonstrated that integrated converters can be reconfigured to generate controlled alternating currents for internal battery heating. This capability allows the system to efficiently warm the battery in subzero conditions without the need for bulky external heating pads. However, this architecture introduces complex electrical and thermal trade-offs. Balachandran *et al.* [143] note that while this integrated topology improves versatility, it can inadvertently increase the internal heat generation within the battery cells during normal traction due to high harmonic current content. One of the main challenges of converter integration is the incompatible operating temperatures of the sub-components. While power electronics typically operate at temperatures exceeding 100°C, batteries require a strictly controlled environment (typically 15–35°C) to preserve safety and lifespan. To investigate these interactions, Zhang *et al.* [140] analysed the steady-state behaviour of a battery-integrated converter utilising a Phase Change Material (PCM) layer. Using a 3D Finite Element Analysis (FEA), they found that while maximising the horizontal thermal conductivity of the cooling medium is essential for dissipating local hotspots, a high vertical conductivity creates a thermal bridge, increasing the heat transfer from the hot converter to the batteries. In a subsequent study, the authors

extended this analysis to transient conditions, demonstrating that the optimal thermal strategy changes with the electrical load [144]. Under low discharge rates, the battery benefits from low vertical conductivity to isolate it from the converter's heat. Conversely, during aggressive discharge, the battery's own Joule heating becomes the dominant factor; in this case, a higher vertical conductivity is required to extract heat from the cells into the PCM. Research on the thermal reliability of power electronics extends beyond battery systems to other critical powertrain components, such as Fuel Cell Electric Vehicles. Alavi *et al.* [145] performed a comparative reliability analysis of non-isolated DC-DC converters for fuel cell applications, demonstrating that 3-level boost topologies significantly reduce the thermal cycling stress on power switches compared to conventional designs, thereby enhancing the overall system lifespan. Many works have also focused on the thermal analysis of Integrated Motor Drives. Wu *et al.* [146] focused on improving motor efficiency by designing a high-power density five-phase Permanent Magnet Synchronous Motor. To ensure the reliability of this compact integration, they conducted a detailed thermal analysis using 3D FEA to verify that component temperatures remained within safe limits. Li *et al.* [147] addressed similar challenges by investigating the thermal behaviour of an Integrated Motor Drive, where heat from motor windings significantly affects the attached power electronics. Recognising that detailed FEA is often too computationally demanding for long transient simulations, they validated a LPTN approach, demonstrating that it provides a necessary balance between predictive accuracy and computational speed. Similarly, Bourgault *et al.* [148] utilised the LPTN methodology on an induction machine to calculate component temperatures and track thermal changes in the stator windings during dynamic operations. This drive for efficient modeling extends to the power converters themselves; Azer *et al.* [149] showed that replacing detailed switching models with State-Space Averaged models coupled to thermal networks significantly reduces simulation time while maintaining accuracy for junction temperatures. Dey *et al.* [139] proposed a 3D thermal network for power transformers by discretising the internal volume into multiple lumped regions. The model was validated against experimental results. Furthermore, Wang and Zhu [150] applied LPTN to Insulated Core Transformers, highlighting its effectiveness in rapidly identifying hotspots within complex segmented core structures during the initial design phase. LPTNs are also extensively used to model full-scale battery modules. Alhanouti *et al.* [151] developed an electro-thermal model for a complete electric vehicle battery pack, their model accounts for temperature effects on battery internal resistance and Open Circuit Voltage (OCV). Damay *et al.* [152] proposed a LPTN for the battery module based on the modeling of the central cell, which was identified as the component subject to the highest thermal stress. Through this approach, they demonstrated the ability to calculate temperatures at strategic points within the battery cell, validating the simulation results through experimental tests. Mannapperuma *et al.* [153] proposed a LPTN model where the parameters are identified using a Genetic Algorithm, which is

fed with real-world drive cycle data rather than standard static characterisation results. By optimising the model against these dynamic profiles, they achieved superior accuracy in predicting the battery pack behaviour under real operating conditions. Saw *et al.* [154] conducted a comparative electro-thermal analysis of cylindrical $LiFePO_4$ cells (specifically 18650 and 38120 formats) using a reduced order model. Their study highlighted a critical trade-off in battery integration: while smaller cells benefit from a lower internal radial thermal resistance, resulting in smaller temperature gradients, they increase the complexity of the pack assembly due to the high number of interconnects required. Conversely, larger cells simplify the mechanical integration and reduce assembly costs, but they exhibit significant internal temperature gradients, thereby imposing stricter requirements on the cooling system. Reiter *et al.* [155] proposed a modular simulation framework for the entire vehicle thermal management system. To balance computational speed with physical accuracy, they employed a hybrid modeling approach: LPTN were used to model the thermal mass and heat transfer of the powertrain components, while a one-dimensional Finite Volume Method was implemented to simulate the fluid transport within the coolant network. Validated against experimental data, this approach shows the capability of reduced order models to simulate complex coupled thermal systems. Beyond characterisation, LPTN models are also effective for real-time control strategies. Murashko *et al.* [156] employed a LPTN based estimator to predict internal cell temperatures in a hybrid thermal management system. This estimation, coupled with a fuzzy logic controller, allowed them to effectively manage thermal inertia and stabilise the battery temperature under dynamic loads. Often, LPTN and analytical models are used in combination with high-order methods, such as FEA and CFD, to derive specific model parameters or boundary conditions. Bhattacharyya *et al.* [157] employed this hybrid approach to calculate the parameters for an air-cooled battery module. These parameters were then used in a reduced-order thermal model, thereby achieving the accuracy of high-fidelity simulations with the computational efficiency required for onboard applications. Torres *et al.* [158] utilised a hybrid methodology for the cooling design of a Silicon Carbide (SiC) integrated motor drive. They developed an analytical thermal model of the multi-layer Printed Circuit Board (PCB), which allowed them to simplify the geometry used in subsequent FEA simulations. By reducing the complexity of the 3D FEA model through analytical pre-calculation, they achieved faster convergence. Furthermore, they employed CFD to estimate the specific intake airflow required to meet the heat transfer coefficients determined by the FEA model. Similarly, Grespan *et al.* [138] adopted a multi-physics modelling strategy for the thermal analysis of an integrated motor drive featuring a six-phase machine and a multilevel converter. They constructed a system-level LPTN model which integrates a dedicated CFD model to characterise the heat transfer within the liquid cooling jacket. The validity of the resulting network was subsequently confirmed by cross-referencing the results with a detailed FEA.

Following this hybrid modelling approach, this chapter presents a thermal model of BIC, where the power electronics are physically integrated with the battery pack on a shared cooling plate. A system-level LPTN was developed to simulate the thermal dynamics of the complete assembly, comprising a power converter and a Li-ion battery pack. To ensure accuracy while maintaining computational efficiency, the model parameters were derived from dedicated sub-models: the heat generation of the battery cells is calculated using a detailed electrochemical model, while the convective heat transfer coefficient of the shared liquid cooling plate is extracted from CFD simulations. This modeling strategy provides a flexible approach that represents a good trade-off between computational effort and accuracy. Moreover, it permits the coupling of the different phenomena scales involved in the full system, enabling a more complete analysis of the full assembly.

3.3. Case Study

The case study analysed in this chapter focuses on the energy storage system of an electric passenger car. The system comprises two battery packs, each rated at 50 kWh, connected in series. Each pack consists of 248 modules arranged in an 8S31P configuration (8 modules in series and 31 modules in parallel). At the module level, each unit contains 26 cells arranged in a 13S2P configuration. The cells are Samsung ICR18650-22P $LiNiMnCoO_2$ cylindrical cells, with a diameter of 18 mm and a length of 65 mm. The nominal cell capacity is 2.15 Ah and the nominal voltage is 3.60 V. The nominal capacity and voltage of a single module and the complete pack are derived from the individual cell parameters. Capacity scales with the number of parallel connections (cells or modules), while voltage scales with the number of series connections. Table 3.1 summarises the data for the energy storage system investigated in this work.

The selected Battery Interfacing Converter (BIC) consists of a single-phase bidirectional two-level/three-level NPC Dual-Active-Bridge (DAB) converter. This topology is commonly used for this function due to its inherent bidirectional power transfer capability, galvanic isolation, high power density, and high efficiency achieved through zero-voltage switching [159–161]. The BIC is composed of High Voltage Switching Cells (HVSCs), Low Voltage Switching Cells (LVSCs), and a high-frequency transformer. The HV side is connected to one of the battery packs, while the LV side connects to the auxiliary 24 V battery. Note that the heat generation of the auxiliary battery is considered negligible; therefore, it is excluded from the thermal analysis to simplify the model. The LV side of the power converter comprises two legs, each with 10 SCs, while the HV side also comprises two legs, each with 4 SCs. Figure 3.1 presents a schematic of the integrated system. It includes the two battery packs (each with a nominal voltage of 374.4 V), the HV and LV converter sides including the transformer, and the 24 V auxiliary battery. The figure also depicts the electric motor and the DC-AC inverter used for traction. While the traction system (motor and DC-AC converter) is

Table 3.1: Analysed battery packs data.

Cell data	
Type	ICR18650-22P
Chemistry	$LiNiMnCoO_2$
Cell diameter [mm]	18.0
Cell height [mm]	65.0
Cell capacity [Ah]	2.15
Cell nom. voltage [V]	3.60
Module data	
Cells in series	13
Cells in parallel	2
Module capacity [Ah]	4.30
Module nom. voltage [V]	46.80
Pack data	
Modules in series	8
Modules in parallel	31
Pack capacity [Ah]	133.30
Pack nom. voltage [V]	374.40

extensively analysed in the PhD thesis by Grespan [132], the present work focuses specifically on the integration of the battery pack with the BIC. Both the battery packs and the power converter are cooled by a shared aluminium cooling plate. The battery packs are positioned on the top surface of the plate, while the power converter components are mounted on the bottom. Figure 3.2a illustrates the top view of the system, highlighting the positioning of the power converter components (located on the opposite face) and the cooling channels path within the cooling plate. The proposed aluminium cooling plate features two parallel rectangular serpentines circulating a 50/50 water-glycol mixture. Figure 3.2f depicts the cross-section of a periodic portion of the cooling plate. The channels are 35 mm wide and 5 mm high, with a centreline distance of 245 mm between adjacent channels. The material thickness separating the channel from the top and bottom surfaces of the plate is 3 mm. Each rectangle in Figure 3.2a represents a single module containing 2 rows of 13 cells; the specific dimensions are detailed in Figure 3.2g. With each pack featuring 248 modules in an 8S31P configuration, the total system comprises 12 896 cells arranged in a lattice of 62 rows and 8 columns of modules. The power converter is strategically placed beneath the cooling plate inlets. This arrangement leverages the maximum temperature difference with the coolant, as the fluid temperature rises as it flows through the plate. Figures 3.2c, 3.2e, and 3.2d detail the dimensions of a single HV leg, LV leg, and the high-frequency transformer, respectively.

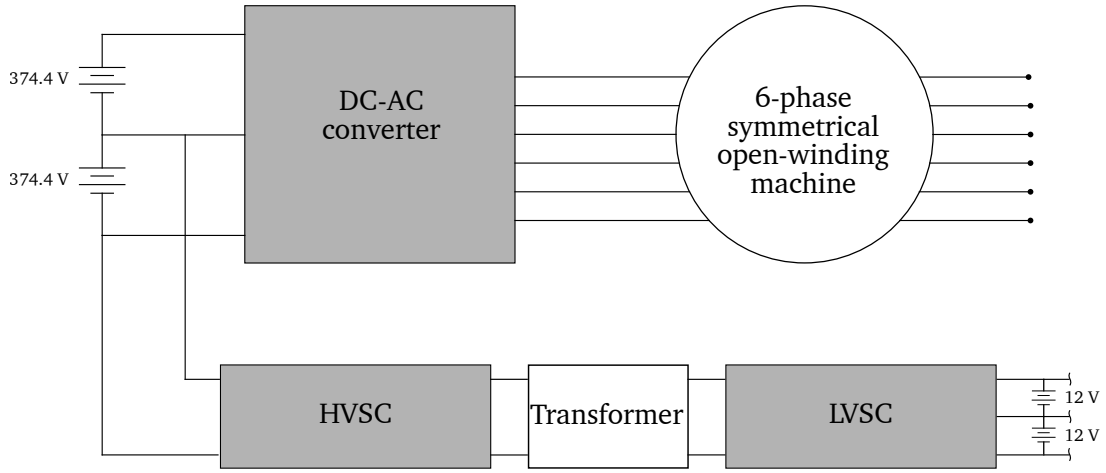
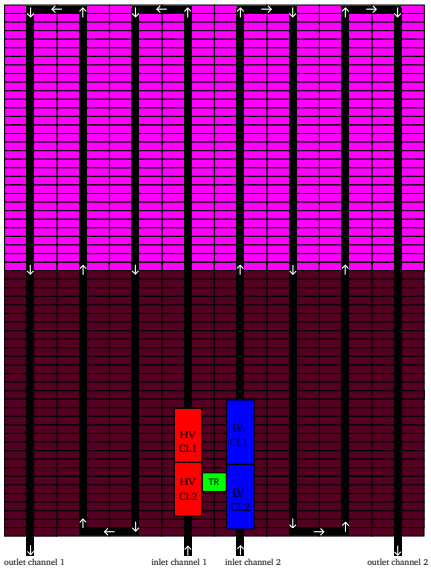


Figure 3.1: Schematic of the integrated system.

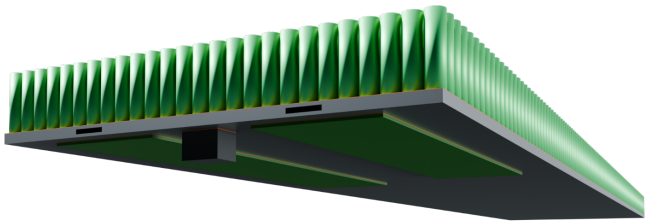
Finally, Figure 3.2b provides a 3D rendering of a periodic portion of the plate, showing the cell stack above and the power converter components below. Between the busbars (in contact with cell terminals) and the top surface of the cooling plate, a 2 mm thermally conductive pad is interposed. This thermal interface material is used to maximise the contact area and increase heat transfer. An identical thermal pad is applied between the bottom surface of the cooling plate and the power converter components.

3.4. Methods

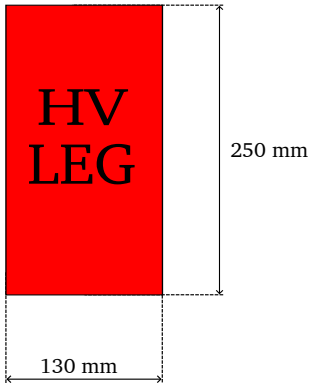
The case study is simulated using the unsteady LPTN solver described in Chapter 2. A CFD analysis was performed to assess the characteristic heat transfer and friction factor of the cooling plate. Due to the large number of battery cells, the LPTN discretises the battery pack at the module level by utilising a scaling law between a single cell and a single module. The heat generation of each cell is derived from a dedicated electro-thermal model based on data from experimental measurements. The losses of each switching cell are also derived from both experimental data and simulations. The model is then applied to simulate the thermal behaviour of the system under a Worldwide Harmonised Light Vehicles Test Procedure (WLTP) drive cycle, incorporating a controller designed to balance the State Of Charge (SOC) between the two packs. Since heat transfer within the cells is predominantly axial, heat transfer in other directions between cells is neglected to simplify the model and reduce computational cost. Finally, to maintain an inlet coolant temperature of 20°C, it is assumed that the vehicle is equipped with an auxiliary heat exchanger (such as a chiller) which cools the coolant entering the cooling plate. In the present work, physical properties are considered constant, the coolant flow is assumed to be incompressible, and radiative heat transfer is neglected, as heat transfer occurs mainly via convection.



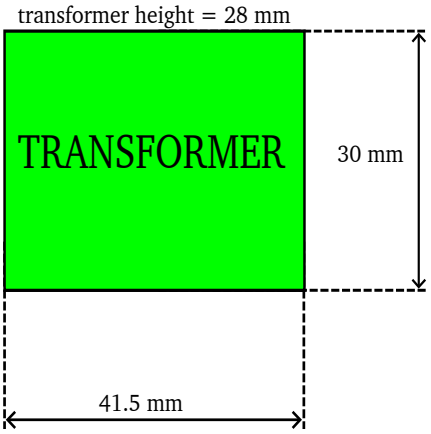
(a) 2D scheme of the integrated system.



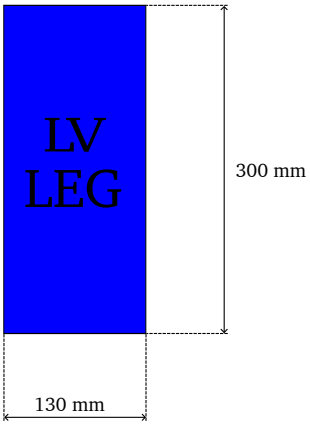
(b) 3D rendering of one portion of the cooling plate with the cell stack above and power converter below.



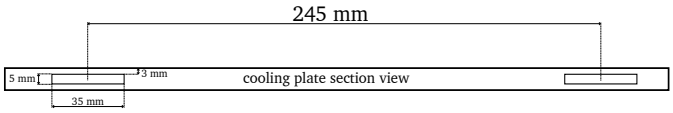
(c) HV leg dimensions.



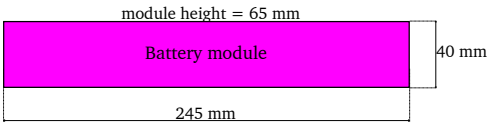
(d) Transformer dimensions.



(e) LV leg dimensions.



(f) Section view of a portion of cooling plate.



(g) Battery module dimensions.

Figure 3.2: Scheme of the analysed system.

3.4.1. Lumped Parameter Thermal Network

As previously mentioned, each module is represented by a single node; therefore, it is assumed that there are no temperature gradients within the module. Consequently, the temperature at the centre of the battery is considered equal to the temperature at the terminals. This hypothesis is considered valid because the thermal conductivity of batteries in the axial direction is typically two orders of magnitude higher than in the radial direction [162]. Figure 3.3 illustrates the thermal network of the batteries and the cooling plate. Figure 3.3a is the side view, while Figure 3.3b is the top view. Following the path of the coolant, the network is composed of a series of vertical sectors. For each module, there is a corresponding thermal resistance to model the contact resistance between the cell terminals and the busbars, followed by a conductive resistance to model the thermal pad between the busbars and the top surface of the plate. Each module has a corresponding coolant node connected to the top and bottom surfaces of the plate via a convective resistance, which is estimated using a CFD simulation of a periodic channel block. Each coolant node is subsequently connected via an advective resistance to the next coolant node, following the direction of the flow. The advective resistances follow the direction of the heat transfer within the coolant, and, in this case, is opposite to the flow direction.

Conductive thermal resistances are calculated according to:

$$R_{\text{th cond}} = \frac{1}{S\lambda}\,, \tag{3.1}$$

where λ is the thermal conductivity and S is the conductive shape factor. For a flat plate, this is calculated as:

$$S_{\text{fp}} = \frac{A}{s}\,, \tag{3.2}$$

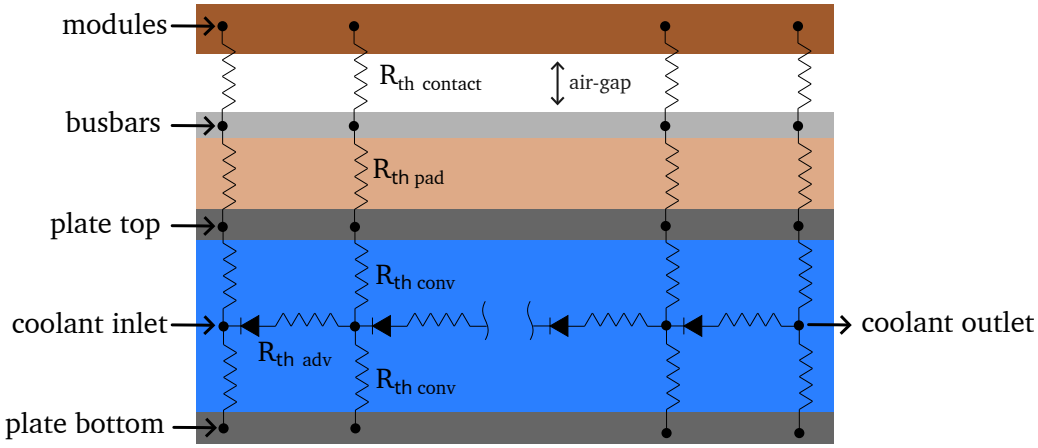
where A is the cross-sectional area and s is the thickness along the heat transfer direction. The contact resistance between the cell terminals and the busbars is calculated as an equivalent conductive resistance using the thickness of the gap and the thermal conductivity of air. In fact, since the air within the gap has zero velocity, it behaves thermally like a solid.

$$R_{\text{th contact}} = \frac{s}{A\lambda_{\text{air}}}\,, \tag{3.3}$$

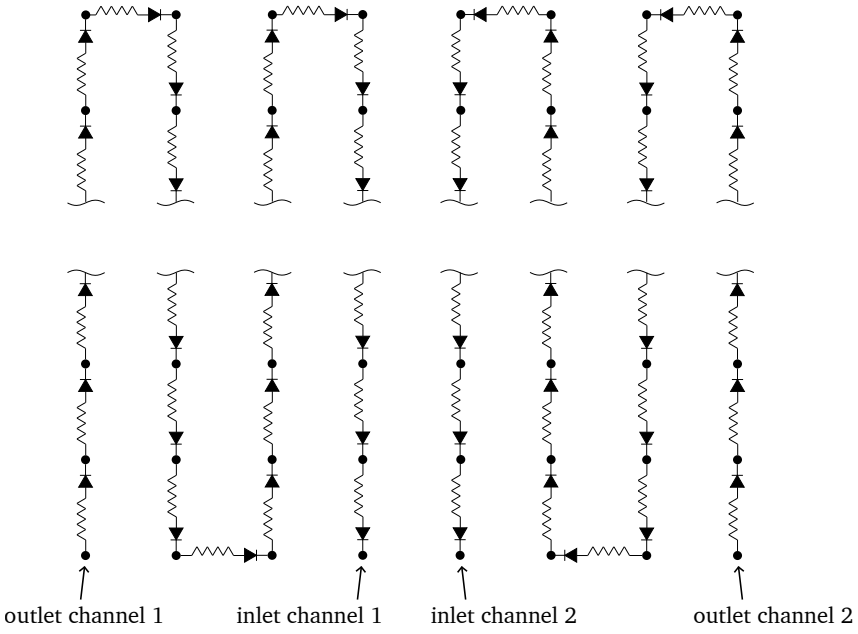
where s is the thickness of the air gap. In this study, an air gap of 0.0026 mm is considered, consistent with the value for copper-copper contact presented in [163]. The convective resistance between coolant nodes and plate surfaces is calculated as:

$$R_{\text{th conv}} = \frac{1}{UA}\,, \tag{3.4}$$

where U is the convective heat transfer coefficient, obtained from a CFD simulation. The



(a) Side view of the modules and cooling plate thermal network (not to scale).



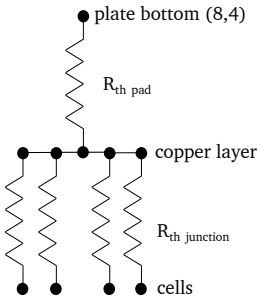
(b) Top view of the modules and cooling plate thermal network.

Figure 3.3: Network representation of the battery modules and cooling plate.

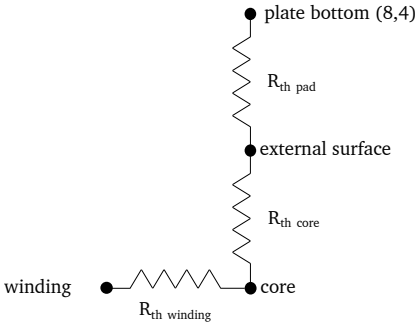
advective resistances are defined only between two fluid nodes and are used to model the temperature rise of the fluid as it flows. In forced convection systems, the direction of heat transfer within the operating fluid is known *a priori*, as it always flows from the region of higher temperature to the region of lower temperature. In this case, heat transfer within the coolant proceeds from the outlet to the inlet, *i.e.*, it follows the reverse path of the fluid. For this reason, advective resistances are modelled as unidirectional branches (see Equation (2.7)), where the direction aligns with the heat transfer (from outlet to inlet). Fixing the direction of these branches prevents errors during the initial iterations of the solver, when the temperature field is far from convergence and the heat direction in these branches could be incorrectly reversed. Advective resistances are calculated as:

$$R_{\text{th adv}} = \frac{1}{\dot{m} c_{\text{p}}}\,, \tag{3.5}$$

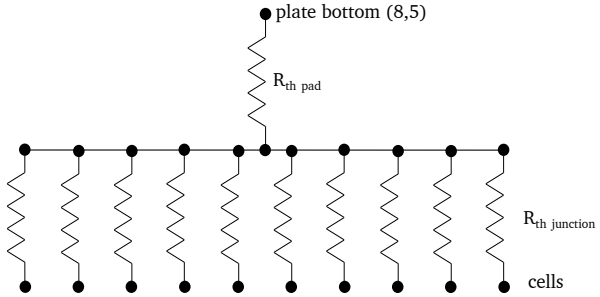
where $\dot{m}$ is the mass flow rate of the coolant and c_{p} is its specific heat capacity at constant pressure. Figure 3.4 shows the thermal network for the power converter components. The network for a single HV leg is shown in Figure 3.4a, where each SC is represented by a node connected to the copper layer node through the junction thermal resistance. The copper layer is then connected to the cooling plate bottom node via the conductive resistance of the thermal pad. The same logic is applied to the modelling of the LV legs, as depicted in Figure 3.4c. The transformer network (3.4b) includes the winding and core nodes linked by the winding-core thermal resistance. The core node is connected to the external surface of the transformer via conductive resistance representing the core's body. The transformer surface node is connected to the bottom surface of the plate through the thermal pad resistance. All the power converter components are connected to the rest of the network by linking them to 2 different cooling plate nodes. The HV side and the transformer are connected to the node representing the bottom plate's surface corresponding to the position of the module at the 8^{th} row from the bottom and the 4^{th} column from the left in Figure 3.2a, while the LV side is connected to the plate's surface node corresponding to the 8^{th} row from the bottom and the 5^{th} column from the left. These nodes were selected as they are located near the geometric centre of the components, making them the most suitable point to assume the heat from the power converter components is lumped. The total losses of each single SC are the sum of the switching losses, due to the non-instantaneous switching between on/off positions, and conduction losses, due to current flowing through the MOSFET channel. Figure 3.5 shows the losses of each SC in a single LV leg (total 5.22 W), derived from a dedicated model. The losses for a single HV leg are 9.5 W, distributed equally among the SCs; therefore, the loss is 2.38 W for each SC. Transformer total losses are 8.6 W, divided into 2.3W for the core and 6.3W for the windings. The losses of transformer components are derived from experimental tests, as are the thermal resistances used for the junction resistance and transformer components. The



(a) Network representation of a single HV leg.



(b) Network representation of the transformer.



(c) Network representation of a single LV leg.

Figure 3.4: Network representation of power converter components.

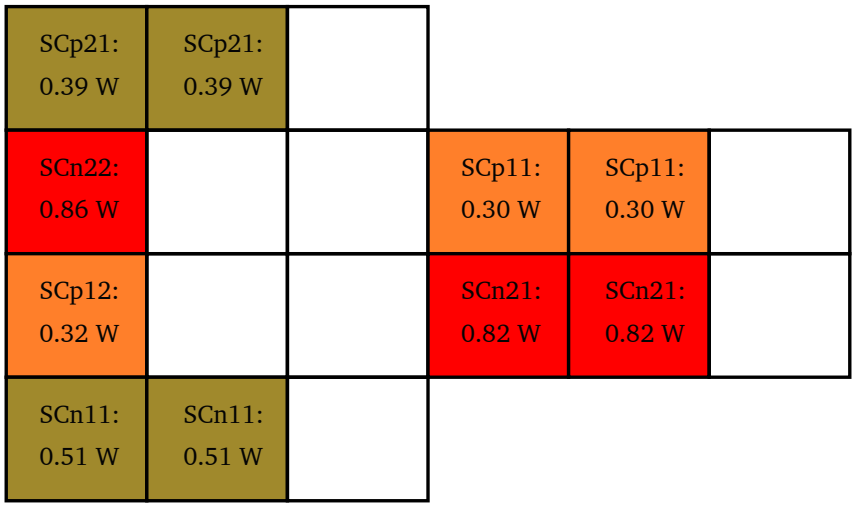


Figure 3.5: LV leg losses.

thermal capacity of the power converter components is neglected because the transient phase of these components is very fast (1 s) compared to the transient phase of the battery packs. Table 3.2 lists the parameters used in the LPTN model. The coolant is a 50/50 water-glycol mixture, which is commonly used in heat transfer applications because it lowers the freezing point and raises the boiling point, although it has lower heat transfer performance compared to pure water. The thermal network used to model the entire system counts 2516 nodes and 2516 branches.

3.4.2. Electro-thermal model

To evaluate the losses in each module, a 3rd order Thevenin model [164] is utilised to calculate the instantaneous voltage, while the Bernardi equation [165] is employed to evaluate the heat generation of the modules. The Thevenin model is an Equivalent Circuit Model (ECM) widely used in this type of application, as it offers a good trade-off

Table 3.2: Simulation parameters.

General data	
Ambient temperature [°C]	20.0
Terminal-busbar air gap [mm]	0.0026
Channel width [mm]	35.0
Channel height [mm]	5.0
Thermal pad thickness [mm]	2.032
Thermal pad λ [$\mathrm{W\,m^{-1}\,K^{-1}}$]	3.0
Plate material λ [$\mathrm{W\,m^{-1}\,K^{-1}}$]	204
Coolant data	
Total mass flow rate [$\mathrm{kg\,s^{-1}}$]	0.34
Single channel mass flow rate [$\mathrm{kg\,s^{-1}}$]	0.17
Coolant specific heat [$\mathrm{J\,kg^{-1}K^{-1}}$]	3319.20
Coolant inlet temperature [°C]	20.0
Coolant density [$\mathrm{kg\,m^{-3}}$]	1068.75
Coolant λ	0.39
Thermal network data	
LV junction R_{th} [$\mathrm{K\,W^{-1}}$]	0.7
HV junction R_{th} [$\mathrm{K\,W^{-1}}$]	0.63
Convective R_{th} [$\mathrm{K\,W^{-1}}$]	0.31
Transf. winding-core R_{th} [$\mathrm{K\,W^{-1}}$]	0.95
Transf. core-ambient R_{th} [$\mathrm{K\,W^{-1}}$]	7.65
Single module thermal capacity [$\mathrm{J\,K^{-1}}$]	142.31

between accuracy, model complexity, and computational time. Other types of battery models exist, such as the simple battery model, which consists only of a voltage source and an internal resistance; consequently, it cannot account for dynamic effects. While this model is computationally inexpensive, it is also the least accurate. Conversely, physics-based models rely on governing equations that describe the internal physical phenomena of batteries, such as electrochemical processes and transport phenomena. These models are the most complex and are typically used for investigating battery performance at the single cell level [166], rather than at the full pack level. The Thevenin model provides a simplified representation of the battery, modelling it as an electrical circuit comprising resistors, capacitors, and voltage sources. Figure 3.6 depicts the equivalent circuit of the Thevenin model employed in this work. The voltage source represents the OCV of the battery, R_0 denotes the internal ohmic resistance, and the RC parallel branches model the dynamic response during charging or discharging due to the diffusion and charge transfer processes occurring within the battery cells. Specifically, the resistors represent the resistance characteristic of transient phenomena, while the capacitors simulate the transient response. The order of the Thevenin model corresponds to the number of RC parallel branches in the circuit; therefore, a first-order model contains only one RC pair. Each RC pair characterises a specific polarisation effect, with its own time constant, where polarisation refers to any deviation of the terminal voltage from the equilibrium potential, and these effects are more pronounced at low temperatures and high currents [167]. The time constant of each RC pair is defined as $\tau_{\mathrm{i}}{=}R_{\mathrm{i}}C_{\mathrm{i}}$. Polarisation reduces the power density of the battery, leading to lower energy conversion efficiency and increased energy losses; additionally, it contributes to battery degradation. Consequently, a higher model order yields greater accuracy in modelling dynamic responses but requires more extensive parameter identification, as the resistance and capacitance of each RC pair must be obtained through experimental analysis. Typically, the first RC pair captures the faster dynamics (characterised by a lower time constant), while each additional RC pair accounts for slower phenomena with progressively increasing time constants. It is worth nothing that the parameters of the equivalent circuit (V_{oc}, R_0, R_{i}, C_{i}) are not constant values; they exhibit a non-linear dependence on the SOC. Consequently, to ensure the accuracy of the model across the entire operating range, these parameters are typically implemented in the solver as two-dimensional look-up tables derived from experimental characterisation. Furthermore, since the parameters evolve with the charge level, the model requires a continuous estimation of the SOC. Figure 3.7 plots the resistance, capacitance, and Open-Circuit Voltage (OCV) values of the equivalent circuit as a function of SOC. These values, derived from repetitive pulse discharge tests, correspond to a single module. The capacitance plots clearly demonstrate that the third RC pair exhibits much slower dynamics compared to the first two pairs. As the parameters exhibit different scales, Table 3.3 lists the measured values plotted in Figure 3.7 to facilitate interpretation, while Table 3.4 lists the

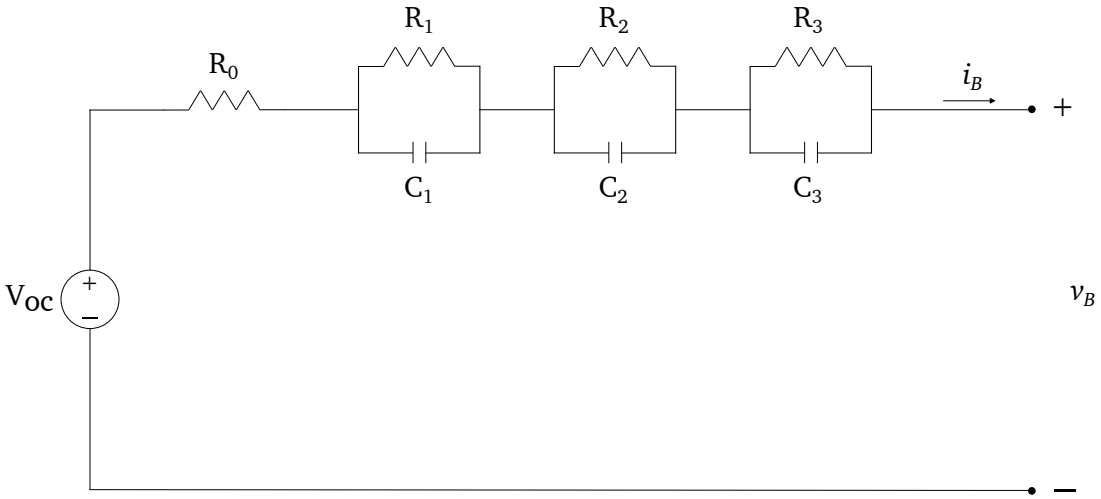
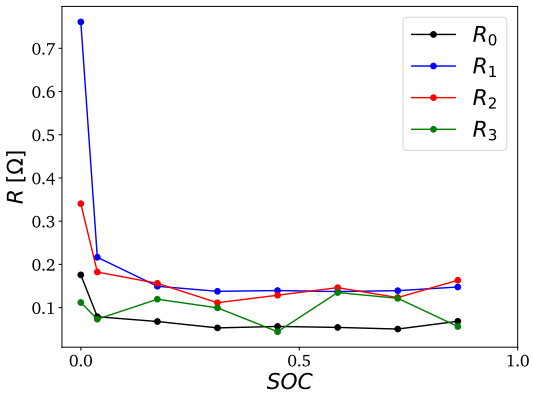
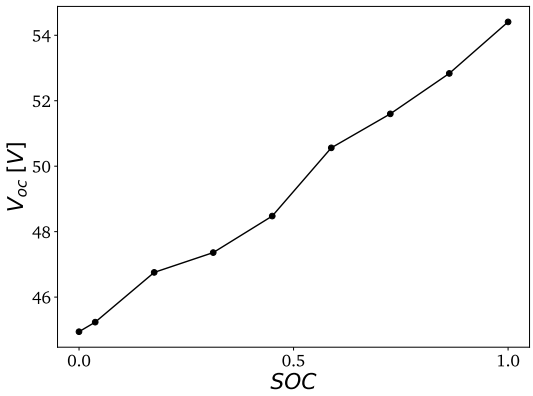


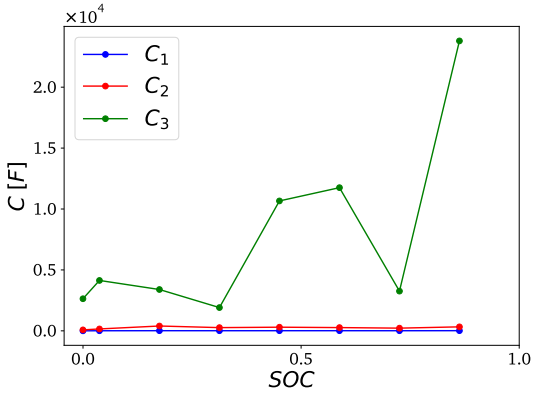
Figure 3.6: Third order Thevenin's model used to model the batteries.



(a) Resistances.



(b) Open-circuit voltage.



(c) Capacitances.

Figure 3.7: ECM parameters identified from the experimental tests.

OCV values. The table highlights the significant difference between the capacitance of the first and second RC pairs. The average characteristic time of the first RC network is 1.48 s, whereas the second RC network is 38.12 s, and the third is 622.59 s.

Table 3.3: Resistances and capacitances values as function of the SOC from experimental analysis.

SOC	0.86	0.73	0.59	0.45	0.31	0.17	0.038	-0.00013
R_0	0.068	0.050	0.054	0.056	0.053	0.068	0.079	0.18
R_1	0.15	0.14	0.14	0.14	0.14	0.15	0.22	0.76
R_2	0.16	0.12	0.15	0.13	0.11	0.16	0.18	0.34
R_3	0.056	0.12	0.13	0.044	0.01	0.12	0.073	0.11
C_1	13.07	9.10	10.91	10.57	9.48	11.41	7.057	1.52
C_2	327.29	221.57	267.17	298.99	264.47	396.62	155.64	79.50
C_3	23798.24	3259.58	11752.39	10663.25	1909.78	3393.0044	4136.57	2635.18

Table 3.4: Open-circuit voltage values as function of the SOC from experimental analysis.

SOC	1.0	0.86	0.73	0.59	0.45	0.31	0.18	0.038	-0.00013
V_{oc}	54.41	52.83	51.60	50.56	48.48	47.36	46.76	45.23	44.94

In the Thevenin model, the instantaneous voltage is given by:

$$v_{\mathrm{B}} = V_{\mathrm{oc}} - i_{\mathrm{B}} R_0 - \sum_{i=1}^{\mathrm{n}} v_i \,, \tag{3.6}$$

where v_{B} is the battery terminal voltage, V_{oc} is the open-circuit voltage, n is the number of RC pairs, and v_i is the voltage across the i-th pair. The voltage evolution of the RC networks is governed by the relation:

$$\frac{dv_i}{dt} = -\frac{v_i}{R_i C_i} + \frac{i_{\mathrm{B}}}{C_i} \,, \tag{3.7}$$

where R_i and C_i are the resistance and capacitance of the i-th RC pair, respectively. By integrating Equation (3.7) over the time interval $\Delta t = t^{k+1} - t^k$, the voltage of the RC networks is calculated as:

$$v_i = \int_{t^k}^{t^{k+1}} \left(-\frac{v_i}{R_i C_i} + \frac{i_{\mathrm{B}}}{C_i} \right) dt \,. \tag{3.8}$$

In the solver, this integral is evaluated using the Implicit Euler method in conjunction with the Newton-Raphson root finding algorithm (refer to Appendix A.3.1). The incorporation of the Newton-Raphson method ensures fast convergence and robust handling of stiff problems. In this work, current is defined as negative during discharge and positive during charge. The SOC is updated at each time step by subtracting the capacity drained during the considered interval from the previous SOC:

$$SOC^{k+1} = SOC^k - \frac{i_{\mathrm{B}} \Delta t}{\mathcal{Q}_{\mathrm{nom}}} \,, \tag{3.9}$$

where $\mathcal{Q}_{\mathrm{nom}}$ is the nominal battery capacity measured in As.

The total heat generated inside a battery is the sum of Joule heating (due to ohmic

resistance), polarisation heating (due to polarisation effects), reaction heating (also referred to as the entropic term), and self-discharge heating [168]. During charging or discharging, self-discharge heating is negligible compared to the other terms [168] and is therefore neglected. Consequently, the battery energy balance reads:

$$\dot{Q}_{\text{gen}} = \dot{Q}_{\text{J}} + \dot{Q}_{\text{p}} + \dot{Q}_{\text{r}} \,, \tag{3.10}$$

where $\dot{Q}_{\text{gen}}$ is the total heat generated within the battery, $\dot{Q}_{\text{J}}$ is the Joule heating, $\dot{Q}_{\text{p}}$ is the polarisation heating, and $\dot{Q}_{\text{r}}$ is the heat generation due to internal reactions (entropic heat). Joule heating is calculated as the product of the battery current and the voltage drop across the ohmic resistance:

$$\dot{Q}_{\text{J}} = i_{\text{B}} v_0 \,, \tag{3.11}$$

similarly, polarisation heating is calculated by considering the sum of the voltage drops across every resistance in the RC parallel branches:

$$\dot{Q}_{\text{p}} = i_{\text{B}} \sum_{i=1}^{\text{n}} v_i \,. \tag{3.12}$$

Finally, the heat due to internal reactions is calculated as:

$$\dot{Q}_{\text{r}} = -i_{\text{B}} T_{\text{B}} \frac{dV_{\text{oc}}}{dT} \,, \tag{3.13}$$

where T_{B} is the battery temperature in Kelvin and $\frac{dV_{\text{oc}}}{dT}$ is the entropic coefficient, defined as the derivative of the open-circuit voltage with respect to temperature. Joule and polarisation heating are always positive during both charge and discharge, whereas entropic heat can be either positive or negative during charge and discharge; this implies that, when the entropic heat is negative, the internal reactions absorb heat rather than generating it. For this reason, it is common to state that Joule and polarisation heating account for irreversible heat generation, while entropic heat represents reversible heat. Therefore, Equation (3.10) becomes:

$$\dot{Q}_{\text{gen}} = i_{\text{B}} v_0 + i_{\text{B}} \sum_{i=1}^{\text{n}} v_i - i_{\text{B}} T_{\text{B}} \frac{dV_{\text{oc}}}{dT} \,. \tag{3.14}$$

By employing Equation (3.6), the energy balance takes its most common form:

$$\dot{Q}_{\text{gen}} = i_{\text{B}} (V_{\text{oc}} - v_{\text{B}}) - i_{\text{B}} T_{\text{B}} \frac{dV_{\text{oc}}}{dT} \,. \tag{3.15}$$

The entropic coefficient exhibits a non-linear dependence on SOC, and its trend can only be evaluated through experimental tests, such as potentiometric or calorimetric methods. Since data regarding the entropic coefficient's trend with SOC for the specific battery considered

is unavailable, this work adopts the entropic coefficient from the calorimetric measurement described in [169], which utilised a battery with the same chemistry, size, and similar capacity. Figure 3.8 shows the measurement results for a single battery. When the battery is completely discharged, the entropic coefficient is negative with a value of $-0.45 \times 10^{-3}\,\mathrm{V\,K^{-1}}$; it then increases following a quasi-linear trend until SOC=0.36. At SOC=0.46, the coefficient changes sign to positive; from this point it remains always positive and its trend is irregular, presenting a peak of $0.2 \times 10^{-3}\,\mathrm{V\,K^{-1}}$ at SOC=0.67. When the battery is completely charged, the entropic coefficient is $0.04 \times 10^{-3}\,\mathrm{V\,K^{-1}}$. These values refer to a single battery cell; to

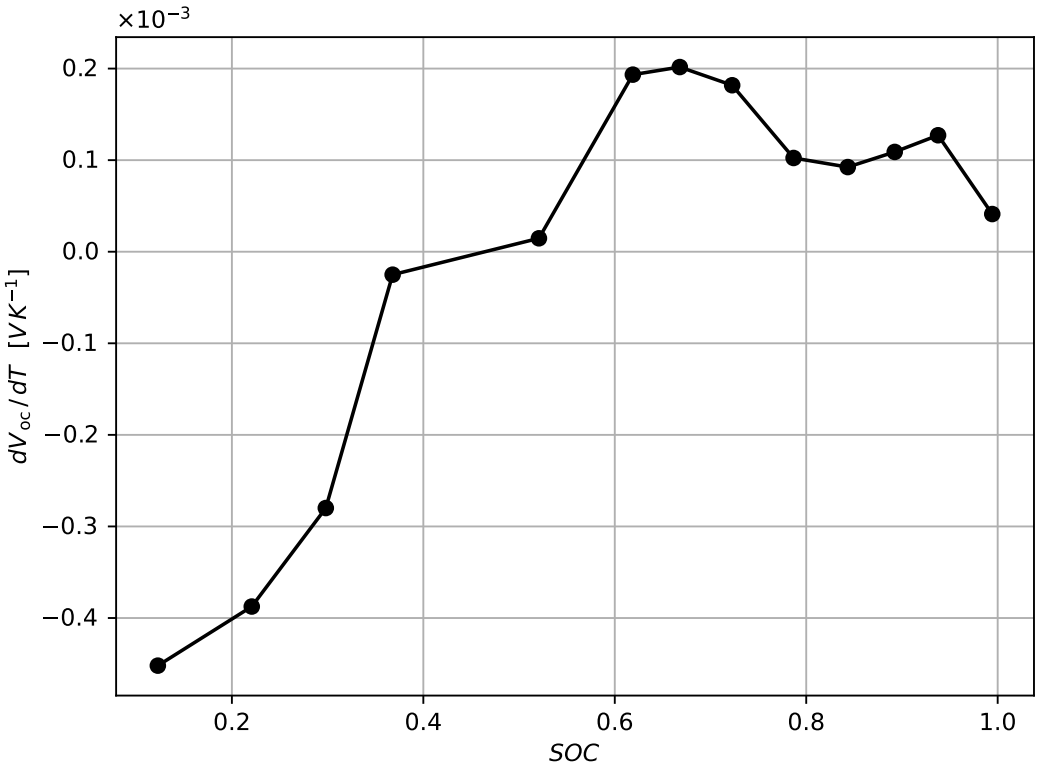


Figure 3.8: Measured entropic coefficient trend with SOC from [169].

scale the entropic coefficient to the module level, it is necessary to multiply by the number of batteries in series within the module.

3.4.3. Control strategy

The control strategy adopted to regulate the current intensity of the two battery packs is explained in detail in the work by Busquets *et al.* [164]; this section focuses on its implementation. Since the two battery packs are connected in series, they would nominally share the same current intensity. However, this can lead to a SOC imbalance, as even identical battery cells exhibit minor differences in voltage, capacity, temperature, and other characteristics due to manufacturing tolerances and operating conditions. In a series connection, the total

capacity is limited by the weakest cell, leading to reduced efficiency. This issue can be mitigated by employing a multibattery-fed multiphase neutral-point-clamped (NPC) DC–AC conversion system. This topology allows for a non-zero current at the neutral point, enabling the two series-connected packs to operate at different current levels. This balancing process incurs no additional losses beyond those already inherent in the power transfer between the battery and other systems, and it allows for SOC balancing even when the packs differ in terms of load, initial SOC, aging, or other parameters. The proposed control strategy is based on a Proportional-Integral (PI) controller. The proportional term generates an output proportional to the error relative to the set-point, while the integral term integrates the error over time to eliminate steady-state error, ensuring that even small deviations are eventually corrected. In general, the controller output $L(t)$ at time t is given by:

$$L(t) = K_{\mathrm{P}} \left(e(t) + \frac{1}{\tau_{\mathrm{I}}} \int_{t_0}^{t} e(t)\, dt \right) , \tag{3.16}$$

where $e(t)$ is the error between the process variable and the set-point, K_{P} is the proportional gain, and τ_{I} is the integral time constant. In this study, the values K_{P}=500 and τ_{I}=20 are used. For the system analysed in this work, the control variable is the SOC imbalance ΔSOC:

$$\Delta SOC = SOC_2 - SOC_1 , \tag{3.17}$$

where SOC_1 denotes the SOC of the battery pack connected to BIC, and SOC_2 denotes the SOC of the other pack. Ideally, the set-point is ΔSOC_{set}=0; therefore, the error is defined as:

$$e(t) = \Delta SOC(t) - \Delta SOC_{set}(t) = \Delta SOC(t) . \tag{3.18}$$

The differential current $i_{\mathrm{diff}}(t)$ required at the neutral point to balance the SOC is obtained from:

$$i_{\mathrm{diff}}(t) = L(t)\, i_{\mathrm{load}}(t)\, \frac{|i_{\mathrm{load}}(t)|}{i_{\mathrm{load}}(t)} , \tag{3.19}$$

where $i_{\mathrm{load}}(t)$ is the current demanded by the load (the WLTP drive cycle in this case). Consequently, the current in the first pack is:

$$i_1(t) = i_{\mathrm{load}}(t) + 0.5\, i_{\mathrm{diff}}(t) - i_{\mathrm{BIC}} , \tag{3.20}$$

and the current in the second pack is:

$$i_2(t) = i_{\mathrm{load}}(t) - 0.5\, i_{\mathrm{diff}}(t) . \tag{3.21}$$

In the above equation i_{BIC} is the steady-state current of the BIC and is calculated as:

$$i_{\mathrm{BIC}} = \frac{P_{\mathrm{BIC}}}{v_{\mathrm{pack}}}\,, \tag{3.22}$$

where P_{BIC}=10 kW is the rated power of the BIC and v_{pack} is the nominal voltage of the battery pack connected to it. Finally, in Equation (3.20), the term i_{BIC} is subtracted to account for the current from the BIC. Equations (3.18)–(3.21) are solved in an iterative procedure until the value of $i_{\mathrm{diff}}(t)$ in the current iteration and that of the previous iteration falls below a specified threshold.

3.4.4. Heat transfer coefficient evaluation

The convective heat transfer coefficient is extracted from a 3D CFD simulation of the cooling plate. In order to reduce the computational cost of the model, only a periodic section of the plate geometry is simulated by imposing periodic flow conditions. This is equivalent to simulating a section in the middle of the plate to obtain the mean heat transfer coefficient. The governing equations for the steady-state flow are the RANS (Reynolds-Averaged Navier-Stokes) equations, which include mass balance, momentum conservation, and thermal energy conservation:

$$\begin{cases} \nabla \cdot \boldsymbol{u} = 0 \\ \nabla \cdot (\,\boldsymbol{u} \otimes \boldsymbol{u}\,) = -\dfrac{1}{\rho}\nabla p + \nabla \cdot [(\nu + \nu_{\mathrm{t}})\,\nabla \boldsymbol{u}] + \sigma \\ \nabla \cdot (\rho\,\boldsymbol{u}\,h) + \nabla \cdot \left(\rho\,\boldsymbol{u}\,\dfrac{|\boldsymbol{u}|^2}{2}\right) = -\nabla \cdot \boldsymbol{q} + \nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u}) \end{cases}\,. \tag{3.23}$$

In this work, $\boldsymbol{u}$ represents the fluid velocity vector, ρ is the fluid density, p is the pressure, ν is the laminar kinematic viscosity, ν_{t} is the turbulent kinematic viscosity, σ is the external momentum source maintaining the flow, h is the specific enthalpy, $\boldsymbol{q}$ is the heat flux vector, and $\boldsymbol{\tau}$ is the viscous stress tensor. The last term of the energy equation represents the heat generated from viscous dissipation, which is negligible compared to the heat generated by the batteries. Therefore, this term is not included in the CFD simulation. The steady-state heat conduction in the aluminium is governed by the Laplace equation:

$$\nabla^2 T = 0\,. \tag{3.24}$$

To account for turbulence effects, the k-ω SST [170] turbulence model is adopted due to its reliability in resolving both low and high Reynolds flows. Figure 3.9 depicts the computational grid of the CFD simulation, where the light-blue cells represent the fluid region and the white cells represent the aluminium region. The mesh is fully structured and

was generated using the OpenFOAM cartesian mesher `blockMesh`. The geometry represents a portion of the cooling plate corresponding to the horizontal extension of a single module, and, to reduce the number of cells (and consequently the computational cost) the horizontal symmetry is exploited, as shown in the 3D view. To enhance accuracy in the boundary layers between the solid and fluid regions, the mesh is gradually refined towards the wall. Once the 2D front mesh is created, the grid is extruded along the normal direction, which also corresponds to the fluid flow direction. The flow is periodic to simulate a fully developed flow scenario; therefore, inlet and outlet effects are neglected, and the fields at the inlet and outlet patches are identical; this periodicity also applies to the temperature at the front and back walls. To maintain the flow between the periodic patches, a momentum source is specified by imposing the mean velocity. In this specific case, the Reynolds number of Re=3000 is assumed, corresponding to a mean streamwise velocity of $0.92\,\mathrm{m\,s^{-1}}$. On

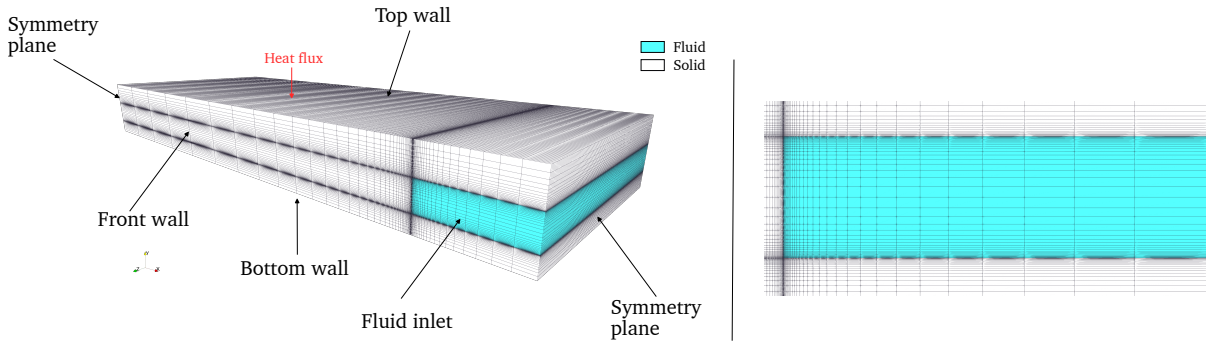


Figure 3.9: 3D view and front view of the CFD computational mesh.

the fluid walls, a no-slip condition is imposed for velocity, and continuity of heat flux and temperature is enforced at the fluid–solid interface. The flow near the walls is resolved using a low Reynolds wall treatment. The convective heat transfer coefficient between the bulk fluid and the plate surfaces is obtained by imposing a heat flux on the top solid wall and adiabatic conditions on the other walls. Finally, a symmetry condition is enforced on the two symmetry planes highlighted in Figure 3.9. Table 3.5 summarises the boundary conditions of the CFD simulation. The problem is solved using the steady-state conjugate heat transfer solver of OpenFOAM (`chtMultiRegionSimpleFoam`), where pressure-velocity coupling is addressed via the SIMPLE algorithm [129]. All advective terms are discretised employing the second-order upwind scheme, while all gradients are solved using the second-order central difference scheme. To assess the influence of the mesh, a mesh sensitivity analysis was performed by comparing the Nusselt number and friction factor across three different meshes: coarse, intermediate, and fine. The coarse mesh is obtained by doubling the cell dimensions of the intermediate mesh in each direction, while the fine mesh is obtained by halving the cell dimensions of the intermediate mesh. Table 3.6 lists the number of cells, the height of the first cell from the wall, the average y^+, the Nusselt number, the percentage difference of the Nusselt number with respect to the coarser grid, the friction factor, and the

Table 3.5: CFD simulation boundary conditions.

Boundary	Velocity	Temperature
Inlet	Periodic	Periodic
Outlet	Periodic	Periodic
Fluid walls	No-slip	Heat and temperature continuity
Top solid wall	-	Imposed heat flux ($q \approx 6\,[\mathrm{kW\,m^{-2}}]$)
Bottom solid wall	-	Adiabatic
Front solid wall	-	Adiabatic
Back solid wall	-	Adiabatic
Symmetry planes	Symmetry	Symmetry

percentage difference of the friction factor with respect to the coarser grid. In all meshes used, $\bar{y}^+ < 1$, confirming that the viscous sub-layer is fully resolved, as expected from the low-Reynolds wall treatment. The Nusselt number shows small variations between the grids, ranging from 7.10 for the intermediate grid to 7.43 for the coarse grid; the greatest variation occurs between the coarse and intermediate grids, with a difference of 4.83% in magnitude. For the friction factor, the differences are even more contained; the greatest difference is observed between the fine and coarse grids (not shown in the table), with a value of 3.4%. These results confirm the reliability of the grids, demonstrating that halving or doubling cell sizes does not have a significant influence on the values of the Nusselt number and friction factor. For the system analysed in the present work, the Nu and f values from the fine grid are considered, since it is the most resolved and theoretically the most accurate mesh. It is worth mentioning that if other conditions need to be tested, the coarse or intermediate grids can be used to save computational cost.

3.5. Results

The described LPTN model was employed to simulate the performance of the system under the WLTP cycle. The WLTP drive cycle is the global standard for assessing the range and energy consumption of BEVs. It lasts 30 minutes and incorporates dynamic driving phases, higher speeds phases, and stricter conditions to provide a more realistic representation of

Table 3.6: Numerical results obtained from the coarse, intermediate, and fine computational grids.

Grid	n_{cells}	$\Delta_{\mathrm{wall}}\,[\mathrm{m}]$	$\bar{y}^+$	Nu	Nu diff.	f	f diff.
Coarse	368×10^3	2×10^{-5}	1.53×10^{-1}	7.46	-	4.97×10^{-2}	-
Intermediate	1×10^6	1×10^{-5}	6.77×10^{-2}	7.10	−4.83%	5.06×10^{-2}	1.8%
Fine	8×10^6	5×10^{-6}	3.88×10^{-2}	7.34	3.27%	5.14×10^{-2}	1.6%

on-road energy usage and vehicle performance. The WLTP cycle used in the present work was repeated twice, for a total duration of 1 hour. Figure 3.10 plots the load current from the test, where a negative current denotes discharge of the batteries, while a positive current indicates charging, which occurs during all phases when the electric motor acts as a generator, such as regenerative braking or, more generally, deceleration events. The numerical solver simulates the entire cycle with a time-step of 0.2 s, and the simulation takes 1.5 hours on a workstation equipped with an i7-2600 CPU at 3.4 GHz. At each step, the corresponding value of the load current is extracted from the data plotted in Figure 3.10; subsequently, the control strategy described in 3.4.3 is applied to split the current between the two packs. Then, for each pack and each module within the pack, the Thevenin model is applied to determine the voltage losses required by Bernardi's equation to evaluate heat generation within the modules. Following this step, the system is solved using the thermal network solver detailed in Section 2.7. Figure 3.11 plots the SOC of the two packs during the simulation, where pack

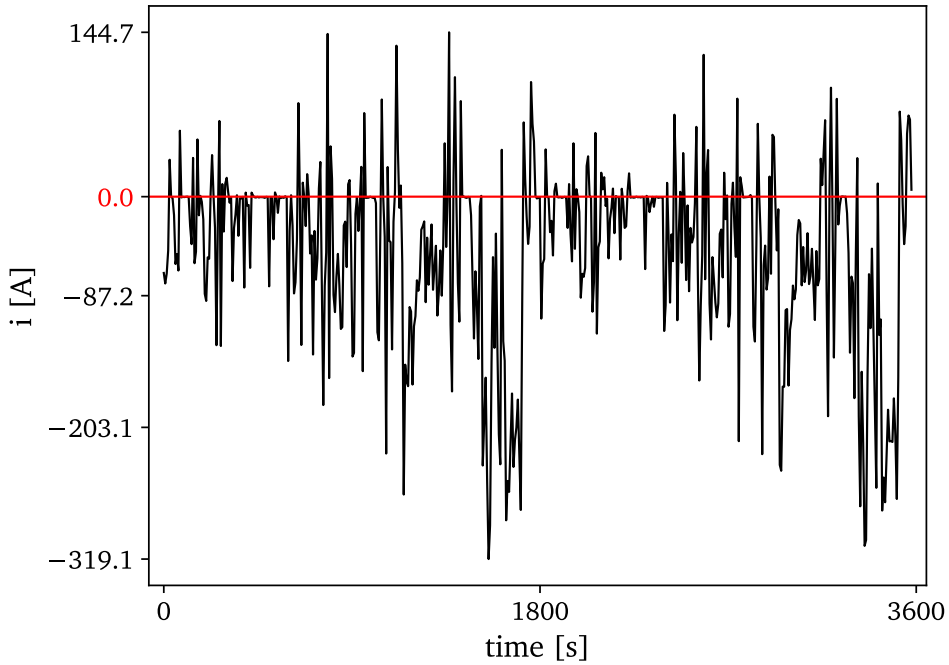


Figure 3.10: Input current from the WLTP cycle test; a negative current indicates discharge of the batteries.

1 is the one connected to the BIC. At the start of the simulation, both packs are fully charged, with SOC=1; they then discharge until SOC=0.54, following the same curve except for negligible differences. This confirms the effective operation of the PI controller in balancing the SOC between the two packs. Figure 3.12 plots the $C\text{-}rate$ of pack 1 (top) and pack 2 (bottom). The $C\text{-}rate$ measures the charge/discharge rate of the energy storage system and is defined as:

$$C\text{-}rate = \frac{i_{\text{pack}}}{\mathcal{Q}_{\text{pack}}\, 3600}\,. \tag{3.25}$$

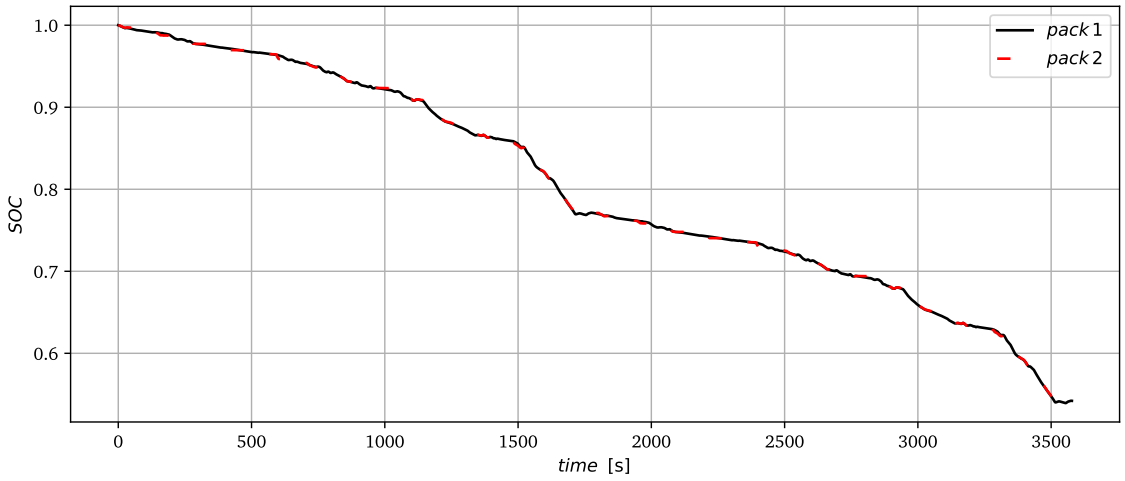


Figure 3.11: SOC trend during the simulation. Pack 1 is the one connected to the BIC.

It is usually measured in h^{-1}; for this reason, the denominator in Equation (3.25) is multiplied by 3600. When $|C\text{-}rate|$=1, it means the storage system is fully charged/discharged in 1 hour; if $|C\text{-}rate|$=0.5, it takes 2 hours; conversely, if $|C\text{-}rate|$=2, the charge/discharge process will be completed in 30 minutes. The $C\text{-}rate$ of the two packs shows a similar trend, although there are some differences due to the control strategy. In particular, the battery pack 2 generally presents negative spikes with higher amplitude with respect to the pack 1, while the latter exhibits positive spikes with a higher charge rate compared to the pack 2. This can be attributed to the fact that pack 1 is connected to the BIC, which drains part of the current; therefore, the controller manages this situation by requiring more discharge current from the pack 2 and charging more the pack 1. The maximum discharge rates for the pack 1 and pack 2 are $-2.60\,h^{-1}$ and $-2.80\,h^{-1}$ respectively, while the maximum charge rates are $1.43\,h^{-1}$ for the pack 1 and $1\,h^{-1}$ for the pack 2. Figure 3.13 shows the heat generated within the battery packs during the simulation; the top plot displays the heat generated from the pack 1, the middle plot refers to the pack 2, and the bottom plot shows the total heat generated, calculated as the sum of the first two. The heat from the pack 2 is generally higher; this is due to the higher $C\text{-}rate$, as also shown in Figure 3.12. The mean value of the heat from the pack 1 is 861.20 W with a peak of ~ 12 kW, while the mean heating from the pack 2 is 1050.16 W with a peak of ~ 14 kW. The spikes in the plots mostly coincide with the negative spikes in Figure 3.12, since both terms in Equation (3.15) depend on the current, and in most instances, the discharge rate is greater in magnitude than the charge rate. The total heat generated within the packs is on average ~ 2 kW, and the maximum value reached is ~ 24 kW. Figure 3.14 plots the heat generated within the packs, distinguishing between irreversible and reversible heat. It is recalled that irreversible heat is the sum of the heat generation due to Joule and polarisation effects and is always positive, whereas reversible heat is due to the internal reactions within the batteries and can be either positive or negative.

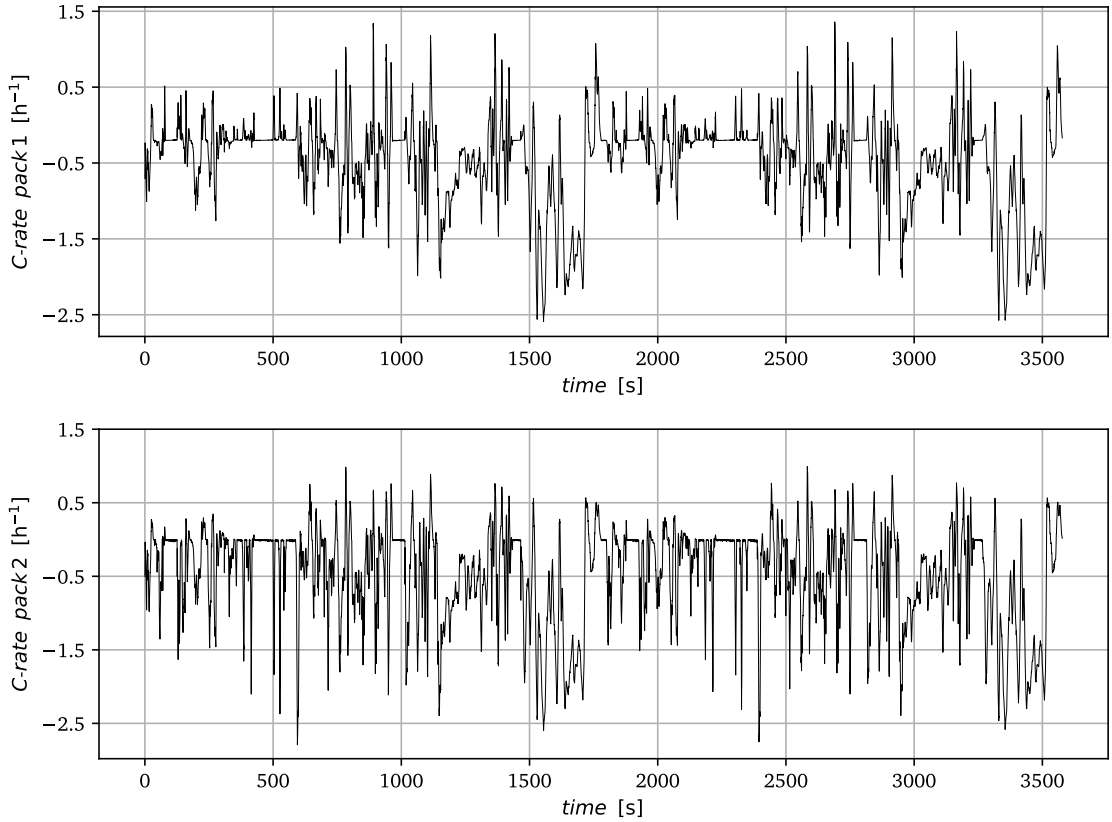


Figure 3.12: *C-rate* of the battery packs during the simulation; top figure refers to the pack 1, bottom figure refers to the pack 2.

In the figure, it is evident that the Joule heating effect is dominant; indeed, the reversible heat is negligible. This behaviour is expected for high C-rates (in magnitude), since the term related to irreversible heat in Bernardi's equation (Equation (3.15)) has a quadratic dependence on the current, while the reversible heat term presents only a linear dependence. In fact, by employing Ohm's law (v=RI), Equation (3.11) becomes:

$$\dot{Q}_{\mathrm{J}} = i_{\mathrm{B}}^2 R_0 \,. \tag{3.26}$$

Furthermore, it can be noted that the reversible heat is negative at certain points, meaning that in those instants the chemical reactions are endothermic, *i.e.* they absorb heat, favouring the battery cooling. In fact, if we consider only Joule heating, the maximum value is $\sim 14\,\mathrm{kW}$ for the pack 1 and $\sim 15\,\mathrm{kW}$ for the pack 2; however, at those instants, the reversible heat is negative, and thus the maximum heating values are lower, as shown in Figure 3.13. The heat generation of the BIC components is constant, and the sum of all the components is $44.34\,\mathrm{W}$. This value is negligible compared to the heating of the batteries; indeed, in Figure 3.15, the temperature difference between the modules interfaced with the LV side (left) and the HV side and transformer (right) is negligible with respect to the preceding modules along

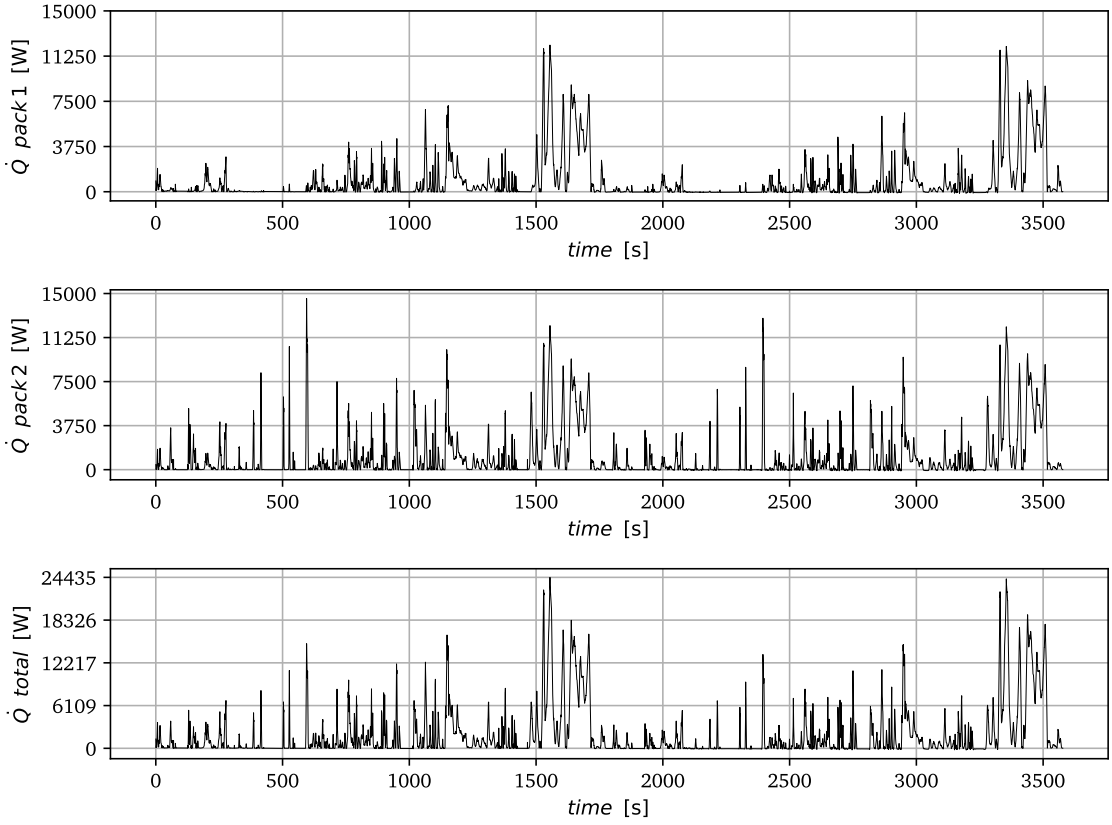


Figure 3.13: Heat generated within the battery packs; top figure refers to the pack 1, centre figure refers to the pack 2, bottom figure is the sum of the first two plots.

the direction of the flow. In the figure legend, the two indices refer to the row and column positions in the grid composed of the modules (see Figure 3.2a), where the row index starts from the bottom and the column index starts from the left. The maximum temperature of those modules is $\sim 29\,^{\circ}\text{C}$, occurring near the middle of the simulation and slightly before the end of the simulation. Thanks to the use of advective resistance (Equation (3.5)) in the network, the heating of the coolant as it flows through the channels is accounted for; this is also reflected in the temperature of the modules. In fact, modules near the coolant inlet are cooled more effectively since the fluid is at its minimum temperature, meaning there is more ΔT available. Moving towards the outlets, the ΔT between the modules and the fluid decreases, and consequently, the heat transfer is lower. This phenomenon is illustrated in Figure 3.16, where the left plot shows the temperature of the first module of the pack 1 and pack 2 encountered by the flow. Since the first module of the pack 1 is located above the inlet of the plate channel, it generally has the minimum temperature among all the modules, with a maximum value of $\sim 29\,^{\circ}\text{C}$. Conversely, the first module of the pack 2 is located downstream, after the coolant has crossed 31 modules, and therefore its temperature is generally higher, considering also that the pack 2 handles higher current values. However, the difference in the maximum temperature is less than $1\,^{\circ}\text{C}$. The plot on the right shows the

temperature of the last modules of both packs. In this case, one might expect the last module of the pack 1 to have a higher temperature since it is situated above the outlet, where the coolant is at its maximum temperature. Despite this, the last module of the pack 2 exhibits a higher temperature for most of the simulated time; this can be attributed to the greater heat generation in the pack 2. Nonetheless, at the two main peaks, the pack 1 shows a slightly higher temperature of 0.58 °C. During the simulated scenario, the mean temperature for the last module of the pack 2 is 23.12 °C, while the mean temperature for the last module of the pack 1 is 22.94 °C. The maximum temperature reached by the modules is 34.84 °C for the last module in the pack 1 and 34.27 °C for the last module in the pack 2. Although these maximum temperatures are near the maximum limit of the operating battery temperature range (15–35 °C), they remain within it. It should be noted that these peak values occur only for short time intervals during the current peaks.

By recording the maximum temperature reached by each module during the simulation time-span, a temperature map of the packs is generated, where each module is coloured according to its maximum reached temperature. The map is presented in Figure 3.17, which clearly demonstrates the effect of the coolant flow. Indeed, following the flow direction, the temperature of the modules increases. This confirms the correct choice of position for the

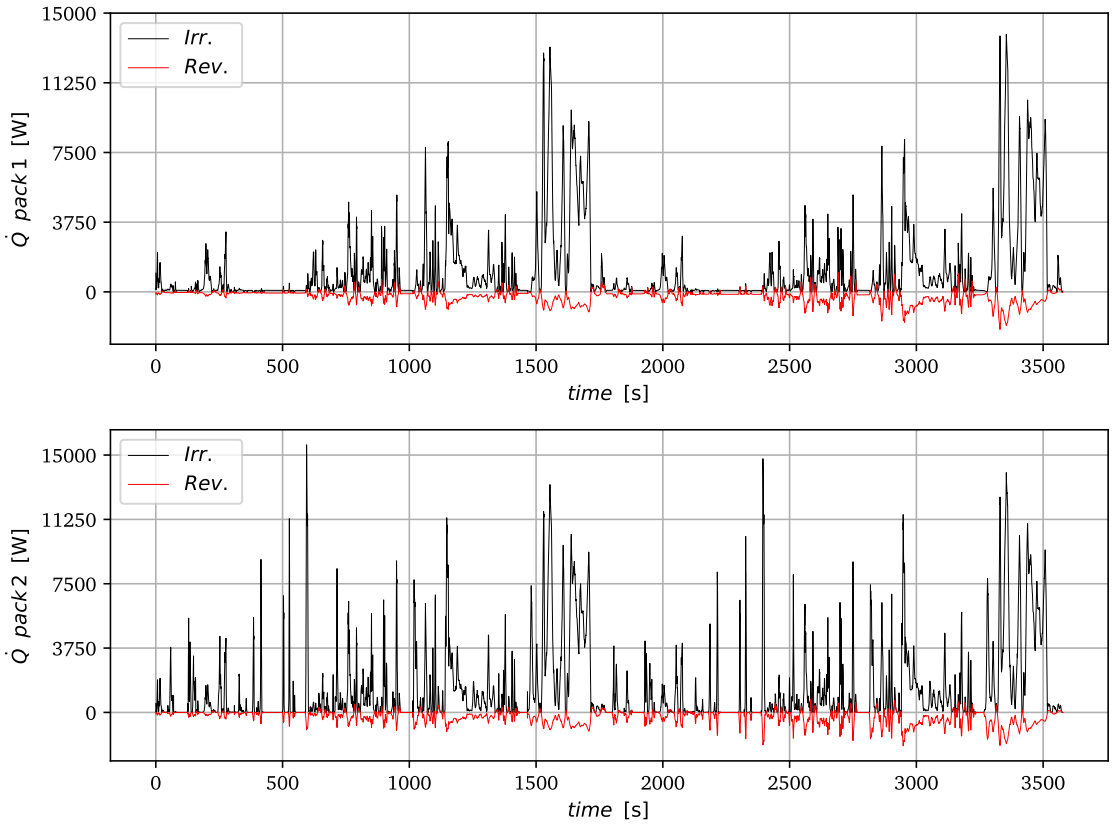


Figure 3.14: Heat generated within the battery packs discriminating between reversible and irreversible heat; top figure refers to the pack 1, bottom figure refers to the pack 2.

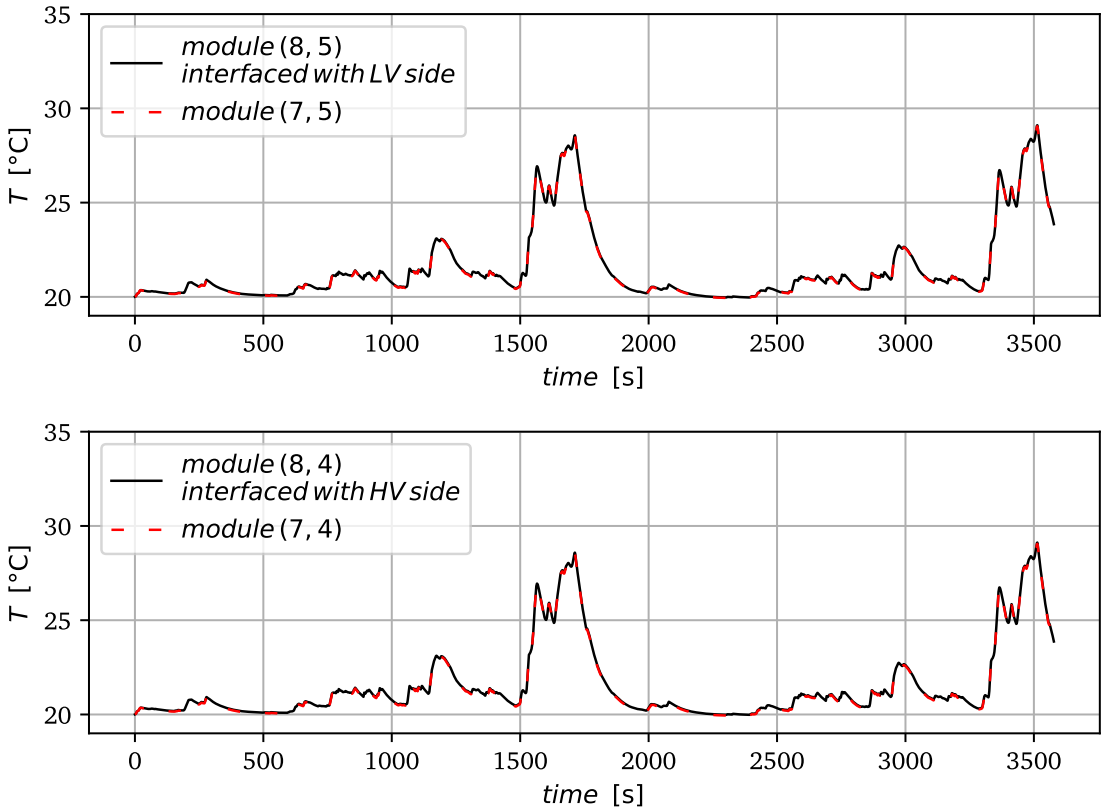


Figure 3.15: Top: temperature trend of the module interfaced with the LV side of the BIC compared with the temperature of the module located one positions upstream along the flow direction. Bottom: temperature trend of the module interfaced with the HV side of the BIC and transformer compared with the temperature of the module located one positions upstream along the flow direction.

BIC components, since they are placed below the region with the lower temperature. The maximum temperature is reached by the modules above the outlet, the ones indicated on the right plot in Figure 3.16. Figure 3.18 plots the maximum temperature difference between the modules of the two packs. To avoid rapid degradation and performance mismatches, the maximum temperature difference within a pack should be kept $< 5\,^{\circ}\text{C}$. Therefore, in designing thermal management systems for battery packs, the focus is not only on absolute temperature values but also on temperature uniformity. In this specific case, the maximum temperature difference occurs in pack 1, between the modules positioned above the inlets and those above the outlets of the cooling channels. During the simulation, this difference reached a maximum value of $6.31\,^{\circ}\text{C}$ near the end of the simulated time. This value could be reduced by increasing the flow rate, using a fluid with a higher thermal capacity, or modifying the channel geometry. It is worth mentioning that for the majority of the time-steps, the maximum temperature difference remains well below the threshold of $5\,^{\circ}\text{C}$. Since the power converter operates at constant power, the temperature of the components does not

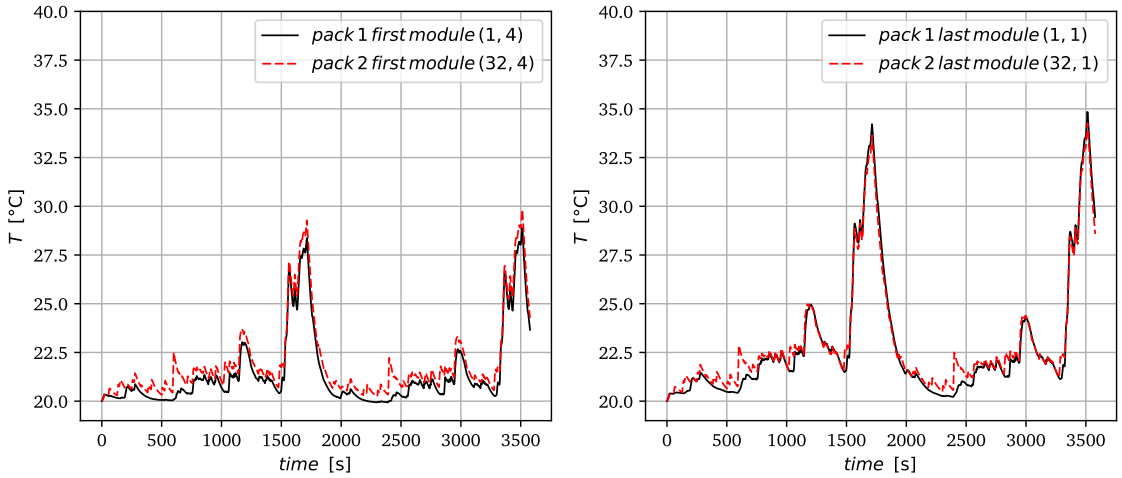


Figure 3.16: Left: temperature trend of the first module of the pack 1 and pack 2 encountered by the coolant flow. Right: temperature trend of the last module of the pack 1 and pack 2 encountered by the coolant flow.

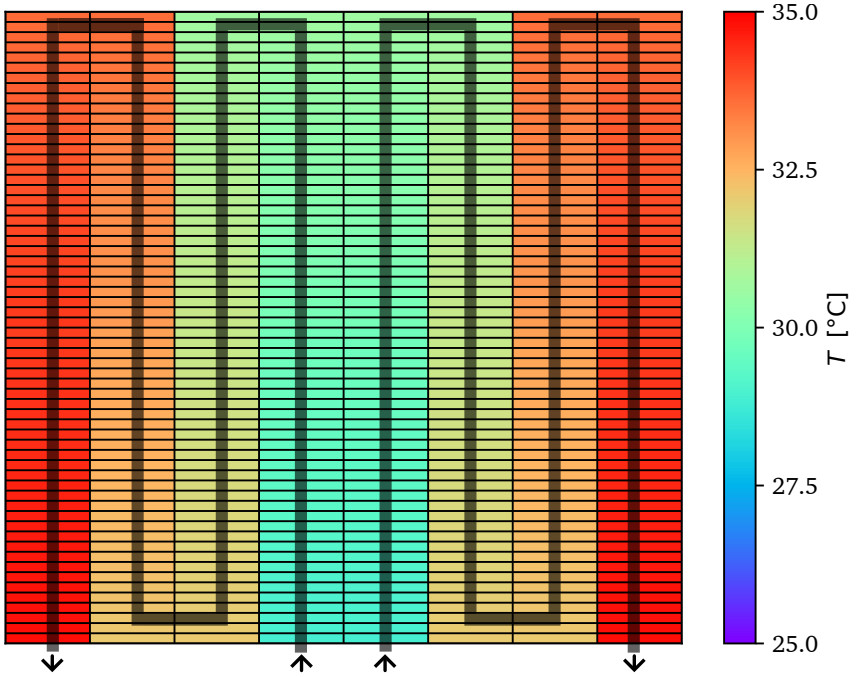


Figure 3.17: Temperature map of the two packs, where each module is coloured according to its maximum reached temperature.

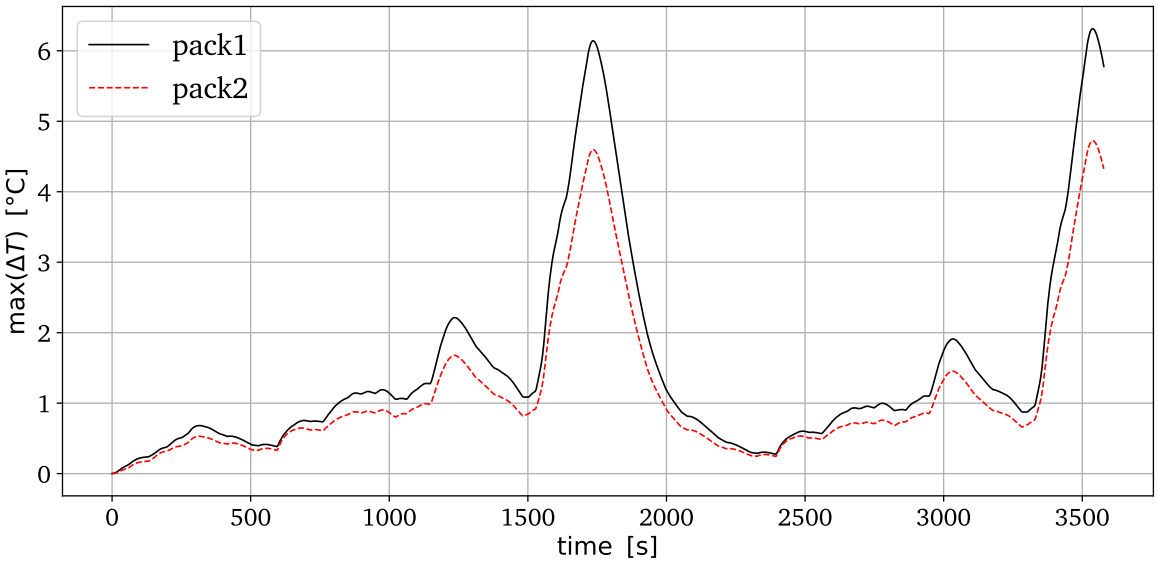


Figure 3.18: Maximum temperature difference between the modules within the same pack. The black line represents pack 1, and the red line represents pack 2.

present significant oscillations, as depicted in Figure 3.19, which plots the temperature of the transformer winding and core nodes, as well as the maximum temperature reached by the SCs on the LV and HV sides. The operating temperature for the switching cells ranges from −40 to 150 °C, while the operating temperature for the high-frequency transformer is usually limited to 125 °C. If these components operate at temperatures higher than the maximum allowed, efficiency decreases, and there is a risk of thermal runaway, which can also lead to thermal runaway in the batteries due to the high temperatures. During the simulation, the maximum temperature reached by the LVSCs is 25.67 °C and the maximum temperature of the HVSCs is 29.38 °C, while the maximum temperatures reached by the transformer core and winding are 99 °C and 105 °C, respectively. Thus, the temperatures of all components are within the operating range. It is recalled that the thermal capacity of the power converter is not accounted for in the simulation, since it is negligible with respect to the thermal capacity of the battery. For this reason, the plot does not exhibit a transient phase. Finally, the circuit pressure losses are estimated using the Darcy-Weisbach relationship (Equation (2.98)) for distributed losses, employing the friction factor extrapolated from the CFD simulation and assuming smooth channels. The concentrated losses due to the turns of the channels are calculated using Equation (2.105). Since the two channels are in parallel, the total pressure drop between the inlet section and the outlet section of the cooling circuit corresponds to the losses of a single channel. In addition, the turns are not sharp as depicted in Figure 3.2a, where turns are drawn with 90° angles for simplicity; instead, they are modelled as turns with a radius of 10 cm. Therefore, since the ratio between the curvature radius and the hydraulic diameter of the channel is > 3.5, the concentrated loss coefficient is K=1.5 [171]. Finally, the value of the pressure losses for the cooling channels is 0.3 bar. This value is

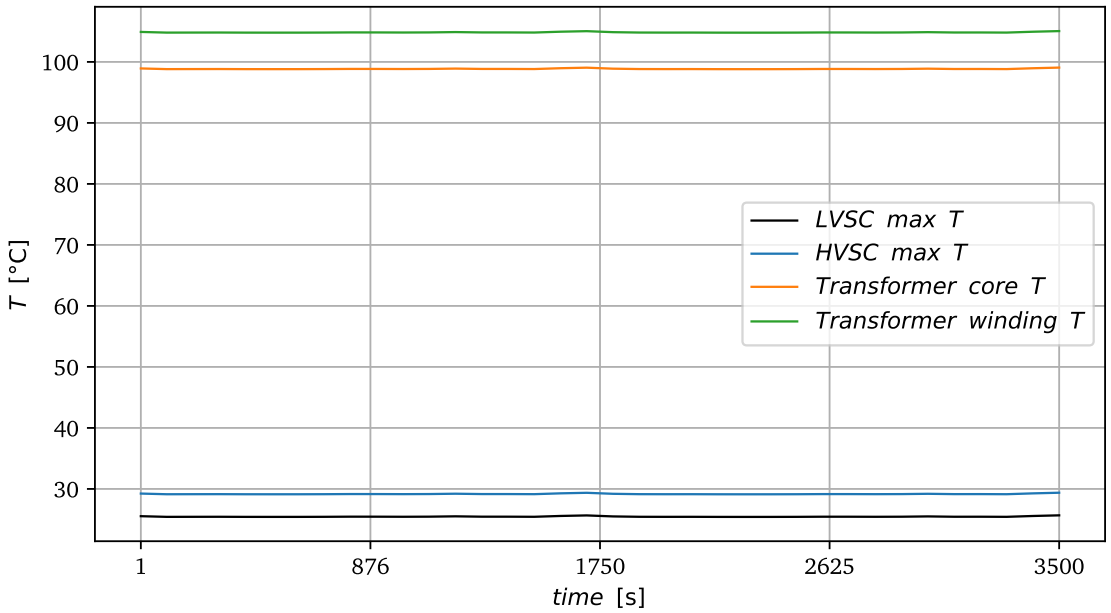


Figure 3.19: Temperature trend of the power converter components.

useful for selecting the pump power required to maintain flow in the system, although in a real-world scenario, the pressure losses of the auxiliary chiller used to recover the fluid at its inlet temperature and the distributed losses along the rest of the circuit must also be considered.

3.6. Concluding remarks

This chapter addressed the thermal management and modelling of an integrated energy storage system designed for an electric passenger vehicle. The case study is a system comprising two battery packs connected in series, each with a rated capacity of 50 kWh. The packs are composed using Samsung ICR18650-22P $LiNiMnCoO_2$ cylindrical cells, organised in a hierarchical structure: cells are arranged in a 13S2P configuration to form modules, which are subsequently connected in an 8S31P configuration to form the complete packs. A BIC is integrated into the energy storage system to convert the high voltage current from the battery packs to the low voltage current required by the auxiliary systems. The thermal management strategy employs a shared aluminium cooling plate, where the battery packs are mounted on the top surface and the power converter components are strategically placed on the bottom surface near the coolant inlets to exploit the minimum temperature of the coolant at the inlet.

The numerical analysis was conducted using a multiscale LPTN model. This methodology effectively bridges the different physical scales of the system, ranging from the electrochemical reactions inside the cells to the macroscopic heat transfer of the full pack and cooling circuit. The heat generation within the modules was evaluated using Bernardi's equation, relying on

voltage drops calculated by a third-order Thevenin equivalent circuit model. The electrical parameters for the Thevenin model were derived from experimental measurements, while the entropic coefficient was obtained from literature studies on cells of identical size and chemistry. Furthermore, the convective heat transfer coefficient for the cooling plate was extracted from a 3D CFD simulation of a periodic channel block under fully developed flow conditions. The reliability of these numerical results was verified through a mesh sensitivity analysis.

The model was employed to simulate the system performance under a repeated WLTP driving cycle, which serves as a realistic dynamic load. Since the BIC is electrically connected to only one of the packs, the discharge is asymmetric between the two packs. For this reason, a PI control strategy is implemented to differentiate the current required from the two packs by the traction motor, aiming to balance the SOC of the packs. The simulation results demonstrated that the heat generated by the power converter components is practically negligible with respect to the heat generation of the energy storage system. All electronic components operated well within their safety limits; the high-frequency transformer and switching cells reached maximum temperatures significantly lower than their rated limits of 125 °C and 150 °C, respectively.

Regarding the battery packs, the cooling system maintains the mean temperature of the pack at safe levels (15–35 °C), with the mean temperature of the hottest module being 23.12 °C. The temperature only approaches 35 °C for two short time intervals, which occur when the load current presents its main peaks. The maximum temperature difference between modules within the same pack is 6.31 °C, occurring between the modules positioned above the inlets and those above the outlets of the cooling channels. However, for most of the simulated time span, the maximum temperature difference in both packs is below 5 °C. In summary, the proposed multiscale modelling approach provides a significant reduction in computational cost with respect to full Computational Fluid Dynamics (CFD) simulations, offering a versatile tool for analysing long transient scenarios and for testing different operating conditions. Future work will focus on optimising the channel geometry of the cooling plate to decrease the maximum temperature reached and improve the temperature uniformity across the modules.

CHAPTER 4

Reduced-order meta-heuristic optimisation of cold plates

4.1. Introduction

The previous chapter analysed the thermal behaviour of an integrated energy storage system cooled by a cooling plate with two parallel serpentine channels. The channel geometry adopted in the cooling plate was fixed *a priori*, and although it ensures safe operating temperatures for the case study investigated, it can be optimised for having better thermal performances. This chapter focuses on a reduced-order meta-heuristic optimisation of the cooling plate, aiming to identify the optimal channels route that minimises both the mean temperature and the temperature non-uniformity of the battery modules.

The optimisation of cooling channels in cold plates, is typically performed by means of topology optimisation relying on density-based methods in Computational Fluid Dynamics (CFD) or Finite Element Method (FEM) frameworks. These approaches allow for the generation of complex, free-form geometries that offer superior thermal performance. However, these methods require high computational resources and often result in complex shapes that are difficult and expensive to manufacture using standard industrial processes. Consequently, there is a need for an optimisation strategy that balances thermal performance with computational efficiency and manufacturability. This chapter presents a novel, reduced-order meta-heuristic optimisation approach for finding the optimal coolant channel paths within a cooling plate, employing Ant Colony Optimisation (ACO); the strengths of the proposed algorithm are its robustness and limited computational demand, as well as the easy manufacturability of the resulting solution.

CFD and FEM based topology optimisation methods are theoretically rigorous but numerically demanding, often involving complex sensitivity analysis and stability issues during

NOMENCLATURE

Latin letters

A	Area [m^2]
b	Random number $\in [0,1]$
b_0	Transition rule min threshold
b_1	Transition rule max threshold
C	Pressure losses coefficient [$kg\,m^{-3}$]
$\boldsymbol{C}$	Coefficient matrix
c_p	Specific heat [$J\,kg^{-1}\,K^{-1}$]
d	distance [m]
D_h	Hydraulic diameter [m]
E	Set of graph edges
e	Number of elitist ant
f	Friction factor
G	Graph
g	Objective function
h	Channel height [mm]
J	Random variable depending on probability distribution
K	Concentrated loss coefficient
L	Total length of a tour [m]
m	Number of ants
$\dot{m}$	Mass flow rate [$kg\,s^{-1}$]
N	Set of graph nodes
$\mathcal{N}$	Feasible next nodes
P	Penalisation
p	Probability
$\mathscr{P}$	Ant tour
Pr	Prandtl number
q	Heat flux [$W\,m^{-2}$]
Q	Deposited pheromone constant
Q_p	Penalisation constant
$\dot{Q}$	Heat [W]
r	Ant rank index
R	Thermal resistance [$K\,W^{-1}$]
Ra	Roughness [m]
Re	Reynolds number
$\boldsymbol{r}$	Known term vector
s	Thickness [m]
SOC	State of charge
T	Temperature [K]
t	Time-step index
U	Convective heat transfer coefficient [$W\,m^{-2}\,K^{-1}$]
$\bar{u}$	Fluid mean stream-wise velocity [$m\,s^{-1}$]
$\dot{V}$	Volumetric flow rate [$m^3\,s^{-1}$]
$\dot{\boldsymbol{V}}$	Volumetric flow rate vector [$m^3\,s^{-1}$]
v	Voltage [V]
w	Channel width [m]

the iterative process. ACO is a meta-heuristic algorithm that is conceptually simpler and easier to implement. Inspired by the natural foraging behaviour of ant colonies, ACO relies on probabilistic rules rather than gradient-based derivatives. In nature, ants initially explore the area surrounding their nest randomly. When an ant locates a food source, it returns to the colony while depositing a pheromone trail. Other ants are attracted to these trails, creating a positive feedback loop that eventually causes the colony to converge on the most efficient path.

In this study, this biological analogy is applied to the thermal domain to provide a novel,

NOMENCLATURE II

Greek letters

α	Pheromone exponent
β	Heuristic function exponent
φ	Global evaporation coefficient
λ	Thermal conductivity $[\mathrm{W\,m^{-1}\,K^{-1}}]$
ξ	Local evaporation coefficient
η	Heuristic function
ν	Laminar kinematic viscosity $[\mathrm{m^2\,s^{-1}}]$
ν_t	Turbulent kinematic viscosity $[\mathrm{m^2\,s^{-1}}]$
ρ	Density $[\mathrm{kg\,m^{-3}}]$
σ	Standard deviation
τ	Pheromone level
$\boldsymbol{\tau}$	Viscous stress tensor [Pa]
ω	Weight

Subscripts

f	fluid
fn	fluid nodes
in	Referred to the global inlet
max	Maximum
min	Minimum
out	Referred to the global outlet
ref	Reference value
s	solid
top	Referred to top surface
tot	Total

reduced-order optimisation approach. The cooling plate is discretised into a grid of nodes, and the algorithm iteratively searches for the most effective fluid paths connecting the inlet to the outlet nodes. The "food source" is represented by the optimal thermal performance, while the "pheromone" is a numerical weight updated based on the quality of the solution.

The thermal evaluation of the generated paths is conducted using a simplified Lumped Parameter Thermal Network (LPTN) rather than full 3D CFD or FEM simulations. This approach accepts that the resulting solutions, inherently simpler with respect to higher-order methods, may not achieve the absolute theoretical performance of free-form higher-order topology optimisation. However, it offers a faster and numerically robust alternative that guarantees manufacturable solutions. Furthermore, the optimal topologies identified by this method can serve as high-quality initial solutions for subsequent, higher-order optimisations.

4.2. Literature survey

As depicted in Table 1.1, for Battery Electric Vehicles (BEVs) with an energy storage capacity $> 62\,\mathrm{kWh}$, liquid solutions are currently employed to manage the temperature of the energy storage system, while for lower capacities forced air cooling is used. Air cooling is easier to implement, lighter in weight, and involves lower costs, but it has a limited cooling capacity. Liquid cooling provides a higher heat transfer coefficient due to the use of coolants with higher thermal conductivity and thermal capacity. Usually the coolant flows inside a cooling plate in contact with the batteries; thus the internal structure of the cooling channels influences the effectiveness of the heat transfer between the solid material of the plate and

the coolant. This internal structure also determines the temperature uniformity of the plate, which consequently impacts the temperature uniformity of the batteries.

Generally, the optimisation of cooling plate channels deals with the shape of the channels, their route, width and length [172]. In this context, topology optimisation offers a method in which all these aspects are varied and optimised by parametrising only one variable: the "porosity" of the material. Specifically, where the porosity is zero, the material is fluid, whereas where the porosity is equal to unity, the material is solid. By optimising the distribution of the porosity field, the best internal topology of the channels is obtained.

Topology optimisation represents a significant methodological advancement beyond traditional parametric design procedures. This technique was originally proposed by Bendsøe and Kikuchi [173] for applications in structural mechanics. In its initial formulation, the optimisation process begins with a design domain fully occupied by solid material, without a pre-defined structure. By evaluating the mechanical stress distribution under applied loads, the algorithm progressively reduces the material density in regions that provide a low contribution to the objective, such as areas with little influence on the overall stiffness of the structure. This process automatically yields an unbiased optimal structure that satisfies the domain constraints.

Although well-established in structural mechanics, the adaptation of this concept to fluid dynamics occurred over a decade later. Borrvall and Petersson [174] were among the first to apply topological optimisation methods to Stokes flows. Their approach adapted the structural mechanics analogy by treating the fluid domain as a porous medium. In this framework, "counter-productive" cells that hinder the optimal flow path are identified and penalised by treating them as solid obstacles. Another significant development in this field was introduced by Othmer [175], who implemented a topology optimisation framework based on a continuous adjoint method, providing a detailed derivation that facilitated the application of these techniques to complex fluid dynamic problems.

Following the developments in adjoint-based optimisation, a significant advancement was presented by Zymaris *et al.* [176] in 2009. They proposed a continuous adjoint formulation for incompressible flows that explicitly computes the adjoint to the Spalart-Allmaras turbulence model. Unlike previous continuous adjoint methods, which relied on the "frozen turbulence" assumption (neglecting variations in turbulent viscosity), this approach derives the exact adjoint equations and boundary conditions for the turbulence model. In the same year, Dede [177] presented topology optimisation for laminar fluid flow which also included heat transfer optimisation, utilising COMSOL. Another key contribution to the development of topology optimisation in forced convection problems is the work by Yoon [178] in 2010, which implemented the laminar Navier-Stokes equations and thermal energy balance within a Finite Element Method (FEM) framework.

From 2010 onwards, research interest in topology optimisation for convective heat

transfer problems grew, leading to the publication of various works. Papoutsis-Kiachagias *et al.* [179] extended the continuous adjoint topology optimisation method to laminar and turbulent flows coupled with heat transfer, demonstrating both 2D and 3D applications of the method. Their objective function sought to minimise pressure losses and maximise the temperature difference between inlet and outlet sections. As the method is gradient-based, precluding the definition of two distinct objective functions, the single objective was defined as a weighted average of the target functions. A key contribution of their work was the derivation of exact adjoint equations for the full system, including the energy equation and the Spalart-Allmaras turbulence model. Additionally, their framework introduced the capability to impose specific volume flow rate constraints at multiple outlets, ensuring hydraulic balancing alongside thermal performance. In an extended work, the same authors [180] demonstrated further applications with constraints on flow swirl and temperature uniformity at the outlet boundaries.

Pietropaoli *et al.* [181] proposed a continuous adjoint optimisation method for three-dimensional Conjugate Heat Transfer (CHT) problems. They applied the method to a 3D square duct, where the optimisation process created complex three-dimensional structures inside the duct to enhance heat transfer. The authors highlighted that while these resulting organic geometries offer superior thermal performance, their complexity makes them primarily suitable for production via Additive Manufacturing (AM). Subramaniam *et al.* [182] investigated the trade-off between heat transfer enhancement and pressure drop reduction in CHT systems using a continuous adjoint framework based on the Finite Volume Method. They generated a Pareto frontier of optimal designs from laminar to moderate Reynolds number flows by testing different weights for the weighted average objective function, validating their findings against solutions derived from Dugast *et al.* [183]. The latter implemented the level-set method within a Lattice Boltzmann Method (LBM) framework, which permitted a clear definition of the solid/fluid interface, thereby avoiding grey scales. They validated the approach with other results from the literature [184].

Gallorini *et al.* [185] addressed the practical implementation of topology optimisation in industrial CHT workflows. They applied a multi-region adjoint solver to the thermal management of electric propulsion PCBs. While the method successfully identified highly efficient cooling channel topologies, the authors highlighted a critical trade-off regarding manufacturability. They observed that the mathematically optimal shapes often feature complex, irregular boundaries that are difficult to manufacture directly. Consequently, a post-processing stage involving surface smoothing and reconstruction is required to generate feasible CAD geometries. Crucially, they reported that this geometric simplification can lead to a deviation from the theoretical optimum, resulting in a slight reduction in thermal performance compared to the raw optimisation results. In another study by the same authors [186] a continuous adjoint topology optimisation framework integrated with a

dynamic Adaptive Mesh Refinement (AMR) strategy is introduced. To address the high computational cost associated with fine grids in topology optimisation, they implemented a hierarchical non-conforming h-refinement technique. This approach automatically refines the mesh in regions with high gradients, specifically at the fluid-solid interfaces, while keeping the mesh coarse elsewhere. The authors demonstrated that this method maintains the accuracy of the sensitivity field and the final topological features while significantly reducing the overall cell count compared to fixed uniform grids.

Following a similar philosophy of enhancing boundary resolution, Galanos *et al.* [187] proposed a topology optimisation framework that combines a cut-cell method with AMR for turbulent CHT problems. Unlike standard density-based methods that rely on penalisation terms, their approach generates a body-fitted grid at the fluid-solid interface in every iteration, allowing for the imposition of exact boundary conditions and the use of wall functions. Furthermore, they introduced a "Think Discrete-Do Continuous" (TDDC) adjoint formulation. This methodology derives the continuous adjoint equations but discretises them using schemes consistent with the primal flow solver, thereby combining the low computational cost of the continuous approach with the gradient accuracy typically associated with discrete adjoints. Yu *et al.* [188] developed a topology optimisation framework for three-dimensional thermal-fluid-structural problems accounting also for thermal expansion. They found that when displacements caused by thermal expansion oppose those from external loads, the optimiser reduces stiffness to exploit this effect, often creating "extruded" 2D-like features and isolated fluid regions that served only to reduce stiffness rather than to cool. Additionally, they demonstrated that imposing stricter displacement constraints inherently compromises cooling performance. Sun *et al.* [189] applied topology optimisation to design mini-channel heat sinks integrated with impact jets. Notably, this is one of the few studies in the field to provide experimental validation of the topology optimisation results. The authors fabricated the complex optimal geometries using AM demonstrating that the experimental measurements of the Nusselt number and pressure drop showed close agreement with the numerical predictions. Liu *et al.* [190] used a density-based approach to investigate the optimal position for inlet and outlet configurations in a micro heat sink. They found that the diagonal configuration offered the highest heat transfer potential (maximum entransy dissipation), while the central inlet-outlet arrangement provided superior synergy between fluid flow and heat transfer. Additionally, they observed that increasing the number of staggered inlets and outlets effectively reduced the maximum surface temperature, albeit with different pressure drop characteristics. Hou *et al.* [191] experimentally investigated the thermal and flow characteristics of a phase change slurry within a topology-optimised cold plate. While the optimised design demonstrated superior thermal performance compared to a standard reference, the authors addressed the challenges associated with its physical realisation. They highlighted that future topology optimisation frameworks should incorporate dimension

constraints to avoid excessively complex features, thereby enhancing manufacturability and reducing the processing costs of the cold plate.

Wu [192] investigated the topology optimisation of a cold plate for a battery module composed of 20 prismatic cells. The objective function is defined as a weighted average of the maximum temperature, the average temperature deviation and the head losses. Compared to a standard cooling plate, the optimised solution increases the Nusselt number by 63% and decreases the pressure drop by 23.9%. Yang *et al.* [193] employed a two-stage topology optimisation procedure to minimise the average temperature, maximise the temperature uniformity and minimise the pressure losses in a liquid cooling plate for a battery module featuring three inlets and three outlets. In the first stage, they executed multiple topology optimisations by varying the positions of the inlets and outlets, with the objective function defined as a weighted average of the aforementioned goals. From the topology-optimised solutions, they generated a response surface for the three objectives depending on the inlet and outlet positions. Subsequently, they applied the NSGA-II algorithm to obtain the Pareto front and utilised the TOPSIS decision method to find the optimal positions for the inlets and outlets. They demonstrated that the topology optimised solution with the optimised inlet and outlet positions achieves better results compared to the topology optimised configuration with standard inlet and outlet positions under various flow rates. Zhong *et al.* [194] optimised a liquid cooling plate for pre-heating batteries. Also in this case, the objective function is defined as a weighted average of three goals: maximise average temperature, maximise temperature uniformity and minimise pressure losses. Comparing the optimised solution to a standard configuration, the average heating rate increased by 7% while maintaining the maximum temperature difference below 5 °C, and the pressure drop decreased by 74.4%.

Dbouk [195] reviewed the engineering design of heat transfer systems using topology optimisation. The author argued that this technique, to be useful, is largely dependent on new manufacturing processes, such as AM. This is because topology optimisation often generates complex, organic shapes that are impossible to produce with traditional manufacturing methods, making 3D printing the only viable way to physically realise these efficient designs. Additionally, the review noted the limited number of fully 3D applications in the existing literature and the absence of methodologies addressing unsteady flow regimes. Fawaz *et al.* [196] reviewed the topology optimisation of heat exchangers. They noted that while AM is often required to produce the complex organic shapes generated by these algorithms, the manufacturing process itself imposes specific constraints, such as minimum length scales and overhang limits. The authors observed that most existing studies do not account for these fabrication limitations during the optimisation process. Consequently, the resulting geometries often require post-processing or simplification to be manufactured, which can reduce the thermal performance compared to the theoretical optimum.

While the majority of topology optimisation studies for CHT problems rely on high-fidelity

CFD or FEM frameworks, alternative approaches based on simplified network representations have been proposed to overcome the substantial computational cost of full-field simulations. In this context, Liu *et al.* [197] developed a hybrid optimisation framework based on a reduced-order network model. In their approach, the cooling channels are represented as a graph of one-dimensional interconnected pipes and junctions. This methodology calculates pressure drop and heat transfer using empirical correlations rather than solving the full Navier-Stokes equations, allowing for rapid design iterations. The authors employed a Genetic Algorithm (GA) to determine the channel topology, coupled with the Method of Moving Asymptotes (MMA) to optimise channel diameters and node positions, and validated the results against 3D CFD simulations.

Other works investigating network-based topology optimisation have focused primarily on district heating. Pizzolato *et al.* [198] applied a network-based topology optimisation to district heating systems. Their objective was to maximise the network's ability to withstand failures (robustness) while keeping the investment cost within a fixed limit. Using a pre-defined grid of possible pipe connections, they demonstrated that introducing loops into the layout significantly increases reliability compared to traditional tree-like structures, albeit at a higher cost. Blommaert *et al.* [199] proposed an adjoint-based optimisation strategy for the topological design of large-scale district heating networks, relying on a network representation of the domain. Their objective is to minimise the investment costs for piping and the operational costs related to pumping, while ensuring that the heat demand of all consumers is met. Egberts *et al.* [200] addressed the issue of uncertainty in heat network design, such as fluctuating heat demands. They proposed a robust optimisation framework based on a GA. This method determines the network topology and pipe diameters by minimising the expected cost (investment and operation) across multiple different demand scenarios. Graph-based topology optimisation has also been applied to structural mechanics. Wang and Tai [201] proposed a method where the structure is represented as a graph, with edges defined by curves of varying thickness. They employed a genetic algorithm to determine the optimal layout, ensuring that the resulting structure remains fully connected and physically meaningful.

In this study, a graph-based discretisation is combined with ACO to optimise the pathway of cooling channels within cold plates. The optimisation process begins with an empty grid of nodes, each of which can be designated as either fluid or solid. The ACO algorithm identifies the optimal continuous path connecting the inlet and outlet nodes by establishing fluid branches. Consequently, the nodes connected by these branches are defined as fluid, while the remaining nodes are defined as solid. Unlike continuous optimisation methods (such as adjoint-based approaches), discrete algorithms like ACO and GA do not rely on gradients to find a new solution, but rather on heuristic rules. This feature avoids the formation of "grey zones", thereby eliminating the need for post-processing methods to

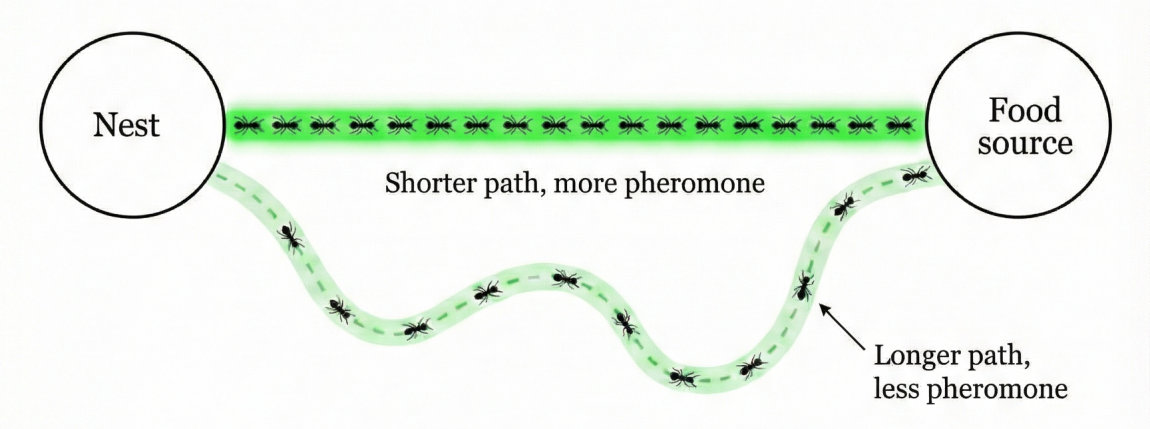


Figure 4.1: Simple scheme of ants behaviour searching food. The ants are attracted by pheromone left by other ants; the shorter the path, the more pheromones are deposited.

sharpen or smooth the obtained solution. This methodology is not intended as a substitute for higher-order methods, but as a computationally efficient alternative for obtaining optimised solutions. These solutions are manufacturable with classic industrial tools and can serve as an initialisation for higher-order optimisation to further refine the final topology. This approach represents a novel application of the ACO algorithm, which, to the best of the authors' knowledge, has previously been employed only for structural problems in the context of layout optimisation [202–205].

4.3. Ant Colony Optimisation

Ant Colony Optimisation (ACO) is a probabilistic meta-heuristic technique for solving hard combinatorial optimisation problems. The algorithm belongs to the class of Swarm Intelligence methods and draws inspiration from the foraging behaviour of real ant colonies. The fundamental principle driving ACO is the capability of biological ants to identify the shortest path between their nest and a food source without using visual cues, but rather by exploiting indirect communication mediated by chemical substances known as pheromones.

The ACO was originally introduced by M. Dorigo in his PhD thesis in 1992 [206] and subsequently formalised in early works on the "Ant System" (AS) [207, 208]. Figure 4.1 is a scheme of ants behaviour in searching food. Ants deposit a pheromone trail on the ground while walking. When an isolated ant encounters a pheromone trail, there is a high probability that it will follow the path; if it does, it reinforces the trail with its own pheromone. This collective behaviour results in a positive feedback loop: the shorter a path, the more frequently it is traversed, leading to a faster accumulation of pheromone compared to longer paths. Consequently, the colony rapidly converges on the optimal trajectory [209].

While originally designed for the Travelling Salesman Problem (TSP), ACO has successfully been adapted to different complex engineering tasks such as structural optimisation, bin packing and cutting stock problems [210], global path planning of mobile robots [211]

and design of aerodynamic shapes [212]. Recently, Ant Colony Optimisation has transitioned from a purely academic research topic to a viable tool for industrial applications. As noted by Dorigo and Stützle [213], several companies have successfully integrated ACO into their operational workflows. This growing industrial adoption is largely due to the algorithm's inherent flexibility and robustness, which allow it to effectively handle the complex, dynamic, and stochastic constraints typical of real-world problems such as routing, logistics, and scheduling.

4.3.1. ACO for Travelling Salesman Problem

To illustrate the mechanics of the meta-heuristic ACO, the Travelling Salesman Problem (TSP) is historically the standard benchmark, as used in the foundational work by Dorigo *et al.* [208] which describes the AS. In the TSP, a set of cities is given, and the objective is to find the shortest closed tour that visits every city exactly once.

The problem is represented as a fully connected graph $G = (N, E)$, where N is the set of cities and E is the set of edges connecting them. The distance between city i and city j is denoted by d_{ij}. In the ACO formulation, a colony of m artificial ants is distributed across the cities. At each time step t, every ant k builds a solution by moving from its current city i to a neighbouring city j using a stochastic transition rule.

The probability p_{ij}^k that ant k chooses to move from city i to city j is determined by two parameters: the pheromone trail τ_{ij} (representing the learned desirability of the edge) and the heuristic information η_{ij} (representing the *a priori* desirability). In the TSP, the heuristic information is typically defined as the inverse of the distance, $\eta_{ij} = 1/d_{ij}$, favoring shorter edges. The rule used to choose the next city is given by the so called *random-proportional rule*:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{if } j \in \mathcal{N}_i^k \\ 0 & \text{otherwise} \end{cases}, \tag{4.1}$$

where $\mathcal{N}_i^k$ is the set of feasible cities, *i.e.* the cities neighbour of city i not yet visited by ant k, and α and β are parameters controlling the relative importance of the pheromone trail versus the heuristic information. In the classical TSP, the graph is fully connected, meaning that every city can be reached from any other; therefore, all cities are considered neighbours. The only constraint is that each city must be visited exactly once.

Once all ants have constructed a complete tour, before time advancing, the pheromone trails are updated. This involves two mechanisms: evaporation and reinforcement. First, a fraction of the existing pheromone evaporates to prevent unbounded accumulation and stagnation, controlled by the evaporation rate $\varphi \in [0, 1]$ which mimics the real evaporation of pheromones. The evaporation is used to speed up the convergence decreasing the probability

of edges with low quality that are not frequently traversed. Second, each ant k deposits a quantity of pheromone $\Delta\tau_{ij}^{k}$ on the edges it has traversed, inversely proportional to the total length of its tour L_k. The global update rule is:

$$\tau_{ij}(t+1) = (1-\varphi)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}\,, \tag{4.2}$$

with

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if ant } k \text{ traversed edge } (i,j) \\ 0 & \text{otherwise} \end{cases}\,, \tag{4.3}$$

where Q is an arbitrarily constant used to set the order of magnitude of the pheromones. This positive feedback loop ensures that shorter tours receive more pheromone, increasing the probability that future ants will select the edges belonging to optimal or near-optimal paths.

In the present, the terms “time”, “generation” and “iteration” are used interchangeably and refer to a single complete tour by all the ants. The pheromones are updated globally between each iteration so that the subsequent generation can exploit the information collected by the previous one.

At the first iteration of the algorithm, an initial pheromone value is required for all the edges, otherwise the probability $p_{ij}^{k}(0)$ is zero for all the branches. In most of the implementations the same initial pheromone value is applied for all the edges, but in some application could be useful initialise the pheromone values with a non uniform distribution based on heuristic information of the graph to avoid ineffective searches during the initial iterations [211].

Despite the innovative nature of the original Ant System, early implementations highlighted several critical limitations. The most significant drawback was the tendency towards stagnation and premature convergence. Due to the strong positive feedback mechanism, if a suboptimal path receives a slightly higher amount of pheromone in the early stages (often due to random chance), it becomes increasingly attractive to subsequent ants. This leads to a scenario where all ants rapidly converge to a single path, often a local optimum rather than the global one, effectively ceasing the exploration of the search space. This behaviour is typical when there is an imbalance between exploration (searching new areas) and exploitation (refining the current best solution). Furthermore, without limits on the pheromone intensity, the difference between high and low pheromone trails could become so large that the probability of selecting an unexplored path drops to near zero [214].

To address these issues and improve performance, several variants were developed to prevent premature convergence and better manage the exploration-exploitation balance. In

the following are reported the most known variants.

4.3.1.1. Ant Colony System (ACS)

Developed by Dorigo and Gambardella [215], this variant introduces a fundamental change to the pheromone update process. ACS uses a specific *local pheromone update* rule applied immediately after an ant crosses an edge (i, j) during the tour construction:

$$\tau_{ij} = (1-\xi)\tau_{ij} + \xi\tau_0 \,, \tag{4.4}$$

where $\xi \in (0,1)$ is the local pheromone decay parameter and τ_0 is the initial pheromone value. This rule reduces the pheromone concentration on visited edges, making them less desirable for subsequent ants in the same iteration, thereby actively encouraging exploration of alternative paths. Additionally, the global offline pheromone update is applied only to the edges belonging to the best tour so far $\mathscr{P}_{\text{best}}$:

$$\tau_{ij}(t+1) = (1-\varphi)\tau_{ij}(t) + \varphi\Delta\tau_{ij} \text{ if edge } (i,j) \in \mathscr{P}_{\text{best}} \,. \tag{4.5}$$

In addition, the *pseudo-random-proportional rule* is also proposed in the ACS, which is governed by a fixed parameter $b_0 \in [0,1]$.

At each step, an ant k located at node i generates a random number b uniformly distributed in $[0,1]$ and compares it to the threshold b_0. The next node j is selected according to the following scheme:

$$j = \begin{cases} \arg\max_{l \in \mathscr{N}_i^k} \{[\tau_{il}]^\alpha [\eta_{il}]^\beta\} & \text{if } b \le b_0 \quad \text{(Exploitation)} \\ J & \text{if } b > b_0 \quad \text{(Exploration)} \end{cases} . \tag{4.6}$$

If $b \le b_0$, the ant behaves deterministically, choosing the neighbour with the highest combined pheromone and heuristic value (exploitation). If $b > b_0$, the ant performs a biased exploration, where J is a random variable selected according to the standard random proportional probability distribution (defined in Equation 4.1).

The parameter b_0 allows the algorithm to be tuned: a value of b_0 close to 1 biases the search towards exploitation (faster convergence, but with the risk to remain trapped in local optima), while a value close to 0 favours probabilistic exploration (avoiding local optima, but slower convergence).

4.3.1.2. Elitist Ant System (EAS)

Proposed by Dorigo *et al.* [208] to address the slow convergence rate often observed in the original Ant System. In standard AS, the pheromone reinforcement is spread across many ants, which can result in a slow differentiation between optimal and sub-optimal paths.

To correct this, EAS allows the "best-so-far" ant (the best solution found since the start of the algorithm) to deposit an additional quantity of pheromone at the end of every iteration, mimicking the guidance of "elite" members of the colony. The primary implication of this modification is a significantly faster convergence speed, as the search is strongly biased towards the best known solution. However, this aggressive exploitation increases the risk of premature convergence, where the algorithm may become trapped in a local optimum before adequately exploring the rest of the search space. The deposited pheromone quantity is calculated from Equation (4.3), except for the edges part of the best tour so far $\mathscr{P}_{\text{best}}$, where a constant gain $e > 1$ (identified as the number of elitist ants in the original paper) it is used to reinforce that tour:

$$\Delta\tau_{ij}^{\text{elite}} = \begin{cases} e\,\dfrac{Q}{L_{\text{best}}} & \text{if edge } (i,j) \in \mathscr{P}_{\text{best}} \\ 0 & \text{otherwise} \end{cases} . \tag{4.7}$$

4.3.1.3. Rank-Based Ant System (AS_{rank})

Introduced by Bullnheimer *et al.* [216] to mitigate the risks associated with the strong bias of pure elitism. While EAS focuses heavily on the single best solution, AS_{rank} proposes a more smooth approach: at each iteration, ants are sorted by solution quality $\mathscr{P}_1 \leq \mathscr{P}_2 \leq \cdots \leq \mathscr{P}_m$. Only the top e ranked ants are allowed to deposit pheromone, where e is the number of elitist ants. The amount of pheromone is weighted according to the ant's rank index r, such that the best paths receive a stronger reinforcement than the lower-ranked ones, therefore Equation (4.2) becomes:

$$\tau_{ij}(t+1) = (1-\varphi)\tau_{ij}(t) + \sum_{r=1}^{e}(e-r)\Delta\tau_{ij}^{r} + \Delta\tau_{ij}^{\text{elite}} , \tag{4.8}$$

This approach prevents a single path from dominating the pheromone field too quickly while ensuring robust exploration.

4.3.1.4. Max-Min Ant System (MMAS)

Proposed by Stützle and Hoos [217], MMAS is widely considered one of the most robust variants. To avoid stagnation, MMAS imposes explicit lower ($\tau_{\min}$) and upper ($\tau_{\max}$) bounds on the pheromone trails. This ensures that the probability of selecting an edge never drops to zero, guaranteeing that all paths remain accessible. Simultaneously, it prevents the pheromone intensity on the best edges from becoming excessive, avoiding premature convergence of the algorithm.

Furthermore, only the best ant (either the iteration-best k_{ib} or the global-best k_{gb}) is

allowed to update the pheromone trails:

$$\tau_{ij}(t+1) = (1-\varphi)\tau_{ij}(t) + \Delta\tau_{ij}^{\text{best}}\,, \tag{4.9}$$

with

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} \dfrac{Q}{L_{\text{best}}} & \text{if edge } (i,j) \in \mathscr{P}_{\text{best}} \\ 0 & \text{otherwise} \end{cases}\,. \tag{4.10}$$

After the update, the pheromone values are bounded to ensure they remain within the feasible range $[\tau_{min}, \tau_{max}]$:

$$\tau_{ij}(t+1) = \max(\tau_{\min}, \min(\tau_{ij}(t+1), \tau_{\max}))\,. \tag{4.11}$$

This strong elitism, balanced by the pheromone limits, allows for a focused search around the best solutions while preserving the possibility of exploring new regions of the search space.

4.3.2. Algorithm steps

To summarise, Figure 4.2 outlines the main steps of the ACO algorithm for the TSP:

i. At the start, all pheromones are initialised. The initialisation can be either uniform (τ_0) or distributed.

ii. The time loop (or generation loop) begins with the first generation of ants.

iii. For each generation, the ants sequentially construct their paths based on the transition rule (Equations (4.1) or (4.6)).

iv. After ant k found a path, the total length is calculated. If this length is shorter with respect to the current best path, the best solution is updated.

v. Local pheromone update (Equation (4.4)) is performed. This step is optional, as introduced in the ACS (Section 4.3.1.1).

vi. If not all ants in the colony have completed their tour, the process returns to step 3.

vii. Once the ant loop is complete, the global pheromones are updated according to the specific model used (Equations (4.2), (4.5), or (4.8)).

viii. If the stopping criteria are met, the optimisation terminates and returns the shortest path. The stopping criteria are typically based on the maximum number of generations or a set number of iterations without improvement in the best solution.

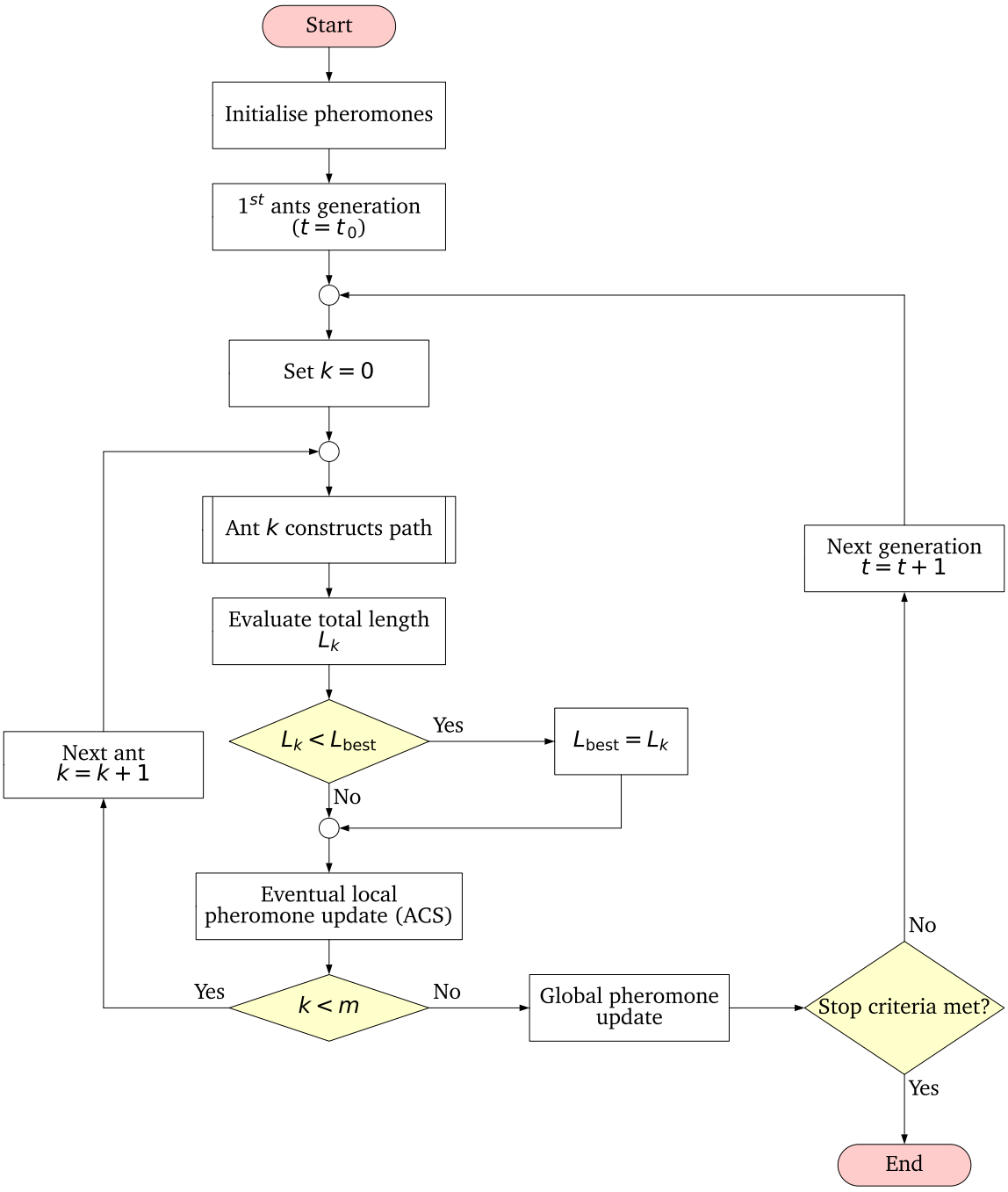


Figure 4.2: Flow chart describing the main steps of ACO for the TSP.

ix. If the stopping criteria are not met, the algorithm proceeds to the next generation of ants.

As illustrated in the flowchart, ACO is a relatively straightforward algorithm that relies on a few fundamental steps. One of the main strengths of ACO, compared to other optimisation methods, is its simplicity of implementation and robustness. While it consistently yields an optimised solution, it is not guaranteed that this solution represents the global minimum. To address this limitation, in addition to developing specific variations, multiple independent runs of the same problem are typically executed, a process that benefits from the algorithm's high potential for parallelisation.

4.4. Proposed algorithm

In this chapter, the ACO methodology is applied to optimise the routing of cooling channels within the cooling plate introduced in Chapter 3, which features two parallel channels with a standard serpentine layout (Figure 3.2a). The goal is to minimise the mean temperature of the plate's top surface (interfaced with the batteries) while maximising temperature uniformity.

Based on a detailed literature review, no similar applications of ACO have emerged; therefore, to the best of the author's knowledge, the proposed algorithm represents a novel application of this method for this specific purpose. Consequently, before describing the algorithm, it is necessary to highlight the differences between the addressed problem and standard ACO applications to justify the adopted assumptions and design choices.

4.4.1. Differences with TSP

Table 4.1 lists the key differences between the classical TSP and the reduced-order cooling channels optimisation.

The first difference lies in the nature of the objective function. For the TSP, the objective is geometrical (minimising the total length of the tour). In contrast, in the cooling problem the objective is thermal, as the goal is to minimise the average temperature of the plate's top surface $\left(\overline{T}_{\text{top}}\right)$ and maximise its temperature uniformity, minimising the temperature standard deviation ($\sigma_{\text{T,top}}$). This distinction affects the cost evaluation: the TSP cost is a summation of edge lengths that is simple to calculate, whereas the cooling problem requires the LPTN solver detailed in Section 2.7 to resolve the system of equations describing the problem. Furthermore, in the TSP, the contribution of a segment to the total cost is known *a priori*. Consequently, defining an effective heuristic function (η) is straightforward, as geometric information, such as the inverse distance, correlates directly with the objective. Conversely, in the cooling problem, the performance of a specific channel segment is difficult to define before the solution is obtained. The actual contribution of a segment is known only after the evaluation as it depends on the resulting temperatures. This implies that geometric

information has a lower correlation with the final thermal performance, making it difficult to guide the ants using only the available information.

Another key difference is the connectivity of the graph used to describe the domain. The TSP graph is fully connected, meaning every city is reachable from any other. Conversely, the cooling problem relies on a more physical representation, where the continuity of the channel must be enforced; an ant can only move to immediate physical neighbours to create a valid channel. This geometric constraint introduces the problem of dead ends situations, where an ant becomes trapped by visited nodes, which cannot be visited a second time, or boundaries before reaching the outlet. While dead ends are impossible in the fully connected TSP graph, they occur in the cooling problem and require specific handling mechanisms.

Additionally, the problem dimension and search space differ. The TSP requires visiting all nodes exactly once, meaning the number of steps is fixed. The cooling problem is a subset selection problem: any node can be either solid or fluid. The optimiser must determine the path sequence and the number of fluid nodes. Consequently, the dimension of the possible solutions is variable.

Finally, the cooling problem allows for the existence of parallel paths, whereas the TSP solution is a single continuous loop. This adds complexity to the algorithm, as it must prevent invalid path intersections and manage flow splitting, where the coolant total mass flow rate is divided among parallel branches.

In summary, the differences stem from two main aspects: the nature of the objective function (geometrical vs. thermal) and the physical constraints of the graph representation. The TSP is a parametrisation of distances, while the cooling problem is a topological representation of a component. These differences increase the complexity of the proposed algorithm, requiring the classical ACO approach to be extended and modified in some features. Thus, to manage this increased complexity, specific assumptions and simplifications are adopted.

4.4.2. Objective function and constraints

Although the optimisation is not gradient-based, hence can handle multiple objectives, here the objective function is defined as a weighted average of the two goals:

$$g = \omega_1 \frac{\overline{T}_{\text{top}} - T_{\text{in}}}{T_{\text{out}} - T_{\text{in}}} + \omega_2 \frac{\sigma_{\text{T,top}}}{\sigma_{\text{T,ref}}}, \tag{4.12}$$

where T_{in} and T_{out} are the coolant temperatures at the inlet and outlet of the cooling plate, while $\sigma_{\text{T,ref}}$ is the reference standard deviation of the temperature between nodes in the plate's top surface. The mean temperature and standard deviation of temperature of the

nodes in the top surface are calculated as:

$$\overline{T}_{\text{top}} = \frac{\sum_i^{N_\text{n}} T_i}{N_\text{n}}, \quad \sigma_{\text{T,top}} = \sqrt{\frac{\sum_i^{N_\text{n}} (T_i - \overline{T}_{\text{top}})^2}{N_\text{n}}}, \tag{4.13}$$

where i iterates on the nodes representing the top surface.

The contributions of the objective function are normalised, since without normalisation, they could have different order of magnitude, and therefore the contribution of one term would be predominant on the other. The coolant outlet temperature is known *a priori*, since

Table 4.1: Comparison of the ACO implementation features for the classical TSP and the cooling channels optimisation problem.

Feature	Classical TSP	Cooling channel optimisation
Objective	Geometric: minimise the total distance.	Thermal: minimise $\overline{T}_{\text{top}}$ and $\sigma_{\text{T,top}}$.
Cost evaluation	Instantaneous: simple summation of static edge weights (distances).	Computationally expensive: requires solving a system of equations.
Segment cost	Independent: the cost of traversing an edge is independent on the final path.	History dependent: the cooling performance of a segment depends on the final path.
Heuristic (η)	Easy definition: typically inverse distance.	Complex to define: the contribution of one branch on the objective is known only after the evaluation.
Graph connectivity	Fully connected: every city connects to every other city.	Physical connected: the paths must be continuous, moving only through neighbouring nodes.
Path constraint	Minimal: dead ends are impossible in a fully connected graph. The only constrain is to visit every city once.	Severe: ants can become trapped in dead ends, without available neighbours.
Path topology	Hamiltonian cycle: the path must visit *every* node.	Simple path: the path does *not* visit every node.
Dimension	Fixed: the number of cities is *known a priori*. The search space is a permutation of a fixed set.	Variable: the channel length is unknown. The algorithm must simultaneously optimise the sequence and the number of visited nodes.
Network topology	Single continuous loop: the solution is strictly a single path.	Multi-path: the solution can consist of multiple parallel channels.

the evaluation is steady-state and the heat flux on the top surface is imposed; therefore:

$$T_{\text{out}} = T_{\text{in}} + \frac{\dot{Q}_{\text{tot}}}{\dot{m}_{\text{tot}}\, c_{\text{p}}}\,, \tag{4.14}$$

where $\dot{Q}_{\text{tot}}$ is the average heat coming from the batteries and $\dot{m}_{\text{tot}}$ is the total mass flow rate flowing in the cooling plate and c_{p} the specific heat at constant pressure of the coolant.

The use of a single objective function simplifies the procedure and aligns it with standard ACO applications. In fact, with the single objective formulation the deposited pheromones can be calculated using directly the rule described in Equation (4.10), simply substituting L with g.

In the algorithm are inserted constraints on the maximum pressure drop between inlet and outlet and the maximum temperature standard deviation. However, these constraints are not strict, since they are treated as penalisation added to the objective function, therefore contributing to the pheromones update. This prevents discarding solutions where these parameters are slightly out of the constraints, considering that the evaluation relies on a reduced-order approach and therefore these values could have some variations with respect to the value resulted on the LPTN. The penalisation terms magnitude depends on how much the value violates the threshold. In this way the contribution of the penalisation term on the deposited pheromone is scaled according to the difference with the constraint. The greater the violation of the constraint, the lower is the deposited pheromone. For the generic quantity ζ, the related penalisation term is:

$$P_{\zeta} = Q_{\text{p},\zeta}\, \frac{|\zeta - \zeta_{\text{max/min}}|}{\zeta_{\text{max/min}}}\,. \tag{4.15}$$

In the penalisation equation, P_{ζ} is the contribution to the total penalisation due to the constraint on ζ, $Q_{\text{p},\zeta}$ is a constant to set the order of magnitude and $\zeta_{\text{max/min}}$ is the constraint on ζ.

In the present algorithm the total penalisation due to the constraint on the maximum temperature standard deviation and maximum pressure drop is:

$$P_{tot} = Q_{\text{p},\sigma_{\text{T}}}\, \frac{|\sigma_T - \sigma_{\text{T,max}}|}{\sigma_{\text{T,max}}} + Q_{\text{p},\Delta\text{p}}\, \frac{|\Delta p - \Delta p_{\text{max}}|}{\Delta p_{\text{max}}}\,. \tag{4.16}$$

Also in this case the contribution of each term is normalised to avoid the dominance of one term on the other.

Additionally to the thermo-fluid dynamics constraints, there is an additional constraint on the number of fluid nodes, similarly to the constraint on the fluid volume used in many higher-order topology optimisation procedures [218, 219]. This constraint is not managed trough penalisation, since it depends only on geometric information, not on the obtained

solution. The constraint is handled by directly discarding the paths that do not respect it, avoiding the waste of ants routes. More details are shown in the explanation of the algorithm for constructing the paths (Section 4.4.5).

4.4.3. Pheromones updating

At the start of the optimisation, all edges are initialised with a uniform pheromone value (τ_0). This value is set to a high level to promote exploration during the first cycle. Subsequently, as the ants generate solutions, the pheromone values are modified according to the updating rules. The proposed algorithm follows the MMAS framework (Section 4.3.1.4) for the global pheromone updating rule and the bounding of pheromone values. Additionally, it incorporates the local updating rule from the ACS (Section 4.3.1.1). The global pheromone update of the MMAS reinforces the edges belonging to the best paths, while the bounding of the maximum and minimum pheromone values prevents premature convergence. This allows for control over the difference in pheromone levels between the most traversed paths and those yet to be explored. In contrast, the local (online) updating rule from the ACS increases exploration by discouraging ants of the same generation from following identical paths.

Therefore, the pheromones are globally updated as in Equation (4.9), where $\Delta\tau_{ij}^{\text{best}}$ is calculated based on the objective function value of the best path:

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} \dfrac{Q}{g_{\text{best}}} & \text{if edge } (i,j) \in \mathscr{P}_{\text{best}} \\ 0 & \text{otherwise} \end{cases} . \tag{4.17}$$

The solution used to calculate the deposited pheromone value (g_{best}) can be either the iteration-best solution or the global-best solution. As discussed in [214], relying solely on the iteration-best solution results in slow convergence, and the algorithm may fail to find high quality solutions for large instances. Conversely, using only the global-best solution can hinder exploration, especially during the initial cycles. Therefore, a mixed strategy is proposed: the pheromone update initially considers only the iteration-best solution. Subsequently, after a defined number of iterations (n_{lb}), a switch between iteration-best and global-best solutions is introduced with a frequency of n_{s} cycles, adding exploitation in the later stages of optimisation.

Following the global update, pheromone values are bounded according to Equation (4.11), where $\tau_{\max}$ is determined as in the original MMAS algorithm:

$$\tau_{\max} = \frac{Q}{(1-\varphi)\, g_{\text{best}}} . \tag{4.18}$$

The minimum bound, $\tau_{\min}$, is calculated as:

$$\tau_{\min} = \frac{\tau_{\max}}{\tau_{\text{ratio}}}\,, \tag{4.19}$$

where τ_{ratio} is the ratio between the maximum and minimum allowed pheromone values. In the proposed algorithm, this is an input parameter that remains constant throughout the optimisation.

The local pheromone update follows the same concept as Equation 4.4. However, instead of the initial pheromone (τ_0), $\tau_{\min}$ is used, and the evaporation factor is the same used in the global update:

$$\tau_{ij} = (1-\varphi)\tau_{ij} + \varphi\tau_{\min}\,. \tag{4.20}$$

This update is applied to each edge of a path found by an ant before the next ant starts its tour. Consequently, the tour of the subsequent ant is influenced by the local pheromone update of the previous ant, which reduces the pheromone value of the traversed path, thereby enhancing the exploration of new branches. The only exception is when τ_{ij}=$\tau_{\min}$; in this case, the pheromone value remains $\tau_{\min}$.

Additionally, to further balance exploration and exploitation, if the global best solution does not improve for n_{imp} iterations, the pheromone levels of all edges are restarted to increase the probability of selecting less traversed edges, as proposed in [214]:

$$\tau_{ij} = \tau_{ij} + b_{\text{res}}(\tau_{\max} - \tau_{ij})\,, \tag{4.21}$$

where b_{res} is a factor in the range $[0,1]$ used to control how much memory of previous cycles is retained. If $b_{\text{res}} = 0$, no change occurs, whereas for $b_{\text{res}} = 1$, all pheromone values are reset to $\tau_{\max}$, resulting in the loss of all previously collected information.

4.4.4. State transition rule

The proposed algorithm uses the *pseudo-random-proportional rule* (Equation (4.6)) from the ACS. As already highlighted, finding a proper function for the heuristic (η) is more challenging in this case compared to the TSP, since the heuristic function must be an instantaneous evaluation relying only on the information available before getting the solution of the current path.

After different preliminary tries (such as maximising the uniformity of the distribution of fluid nodes in the grid, favouring nodes that have fewer fluid neighbours, favouring nodes toward the center and avoiding dead ends), the heuristic function which performed best is the one which tries to maximise the heat transfer area between solid and fluid by avoiding the nodes closer to the outlet, forcing longer paths. This is possible since the inlet and the outlet columns in the grid are known *a priori*: the inlet of a channel is always in the first

column, while the outlet is always in the last column.

To clarify the explanation, Figure 4.3 shows what happens during the choice of the next node to continue the path. Generally, each node has 4 neighbours: one in the top (or north), one in the bottom (or south), one in the left (west) and one in the right (east). The boundary nodes and the corner nodes are an exception since the boundary nodes have only 3 neighbours available, and the corner nodes only 2. As shown in the figure, each node has its own label, and the path is identified by the sequence of labels of the crossed nodes. In this case, the inlet position of the channel is in node 7, then, following the *pseudo-random-proportional rule*, the ant goes to node 8, and repeating the process reaches node 5. From here the same process is repeated again. The available neighbours from node 5 are nodes 2, 4 and 6. The pheromone level of each branch is already stored, so what is missing is only the calculation of η. As already anticipated, the heuristic function aims to choose the neighbour which is more distant from the last column, which is the column where the outlet of the channel is located. Therefore, the heuristic is based on the distance on the grid of the current node and the candidate nodes to the last column. The distance is not calculated on the physical coordinates of the nodes, but on the column index in the grid:

$$\begin{aligned} d_5 &= 1 \\ d_2 &= 1 \\ d_4 &= 2 \\ d_6 &= 0 \end{aligned} \quad . \tag{4.22}$$

In this simple case, the largest distance is 2, since node 4 is in the first column.

At this point η is calculated for each possible neighbour $j \in [2, 4, 6]$:

$$\eta_{5,j} = 1 + \frac{d_j - d_5}{d_{\max}} \, , \tag{4.23}$$

where $d_{\max}$ is the maximum column distance and is simply equal to the total number of columns decreased by 1; in this case $d_{\max}$=2.

Hence:

$$\begin{aligned} &\eta_{5,2} = 1.0 \\ &\eta_{5,4} = 1.5 \\ &\eta_{5,6} = 0.5 \rightarrow \eta_{\min} \quad \text{with} \quad 0.5 < \eta_{\min} < 1.0 \end{aligned} \quad . \tag{4.24}$$

If $\eta_{i,j}$ is zero (or close to zero), the probability of the branch also drops to zero (or close to zero). To avoid this, $\eta_{i,j}$ is bounded to a minimum value $\eta_{\min}$. In this example it is assumed $0.5 < \eta_{\min} < 1.0$, therefore only $\eta_{5,6}$ is bounded.

To show the process of the *pseudo-random-proportional rule* with numbers, the values of

the pheromones of the candidate branches are assumed:

$$\begin{aligned} \tau_{5,2} &= 1.4 \\ \tau_{5,4} &= 2.7 \\ \tau_{5,6} &= 1.9 \end{aligned} \quad , \tag{4.25}$$

and it is also hypothesised that $\eta_{\min}$=0.7, hence $\eta_{5,6}$=0.7. At this point it is possible to calculate the product of the pheromone level and heuristic for each candidate edge:

$$\begin{aligned} \tau_{5,2}\,\eta_{5,2} &= 1.4 \\ \tau_{5,4}\,\eta_{5,4} &= 4.05 \\ \tau_{5,6}\,\eta_{5,6} &= 1.33 \end{aligned} \quad , \tag{4.26}$$

therefore, assuming as example α=1 and β=1, the probability $p_{ij}^{k}(t)$ of each edge according to Equation (4.1) is:

$$\begin{aligned} p_{5,2}^{k}(t) &= 0.21 \\ p_{5,4}^{k}(t) &= 0.60 \\ p_{5,6}^{k}(t) &= 0.19 \end{aligned} \quad . \tag{4.27}$$

Following the *pseudo-random-proportional rule* (Equation (4.6)), a random number

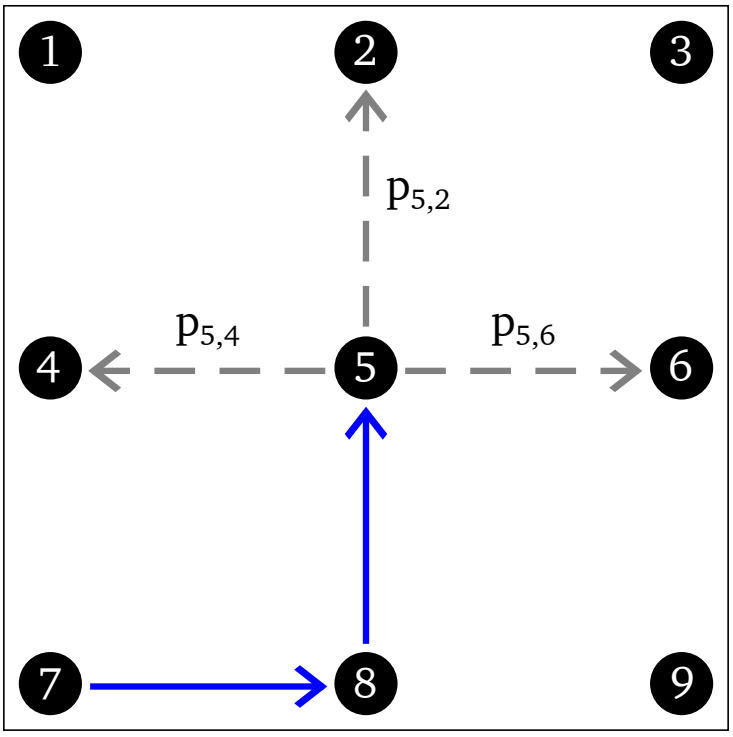


Figure 4.3: Example of an incomplete path in a 3 × 3 grid. The blue arrows represent the current path; the grey dashed arrows indicate the available edges for continuation.

$b \in [0,1]$ is generated. If $b \leq b_0$, the edge with higher probability, $(5,4)$ in this case, is chosen. In the opposite case, the next node is chosen with the *roulette wheel* method, that is a stochastic method based on the probability of the candidates. Therefore it has both aspects, randomness and proportionality. In the *roulette wheel* method the candidates are sorted from the most probable to the least probable, and for each of them the cumulative probability $cf_{i,j}$ is calculated:

$$\begin{aligned} cf_{5,4} &= p^k_{5,4}(t) + p^k_{5,2}(t) + p^k_{5,6}(t) = 1.0 \\ cf_{5,2} &= p^k_{5,2}(t) + p^k_{5,6}(t) = 0.40 \\ cf_{5,6} &= p^k_{5,6}(t) = 0.19 \end{aligned} \quad ; \tag{4.28}$$

an additional random number $b' \in [0,1]$ is generated, and based on the range in which the generated number falls, the next node j is chosen. The ranges are identified by the cumulative sums $\big([cf_{5,4}, cf_{5,2}), [cf_{5,2}, cf_{5,6}), [cf_{5,6}, 0]\big)$:

$$\begin{aligned} j = 4 \quad &\text{if} \quad 0.40 < b' \leq 1.0 \\ j = 2 \quad &\text{if} \quad 0.19 < b' \leq 0.40 \\ j = 6 \quad &\text{if} \quad b' \leq 0.19 \end{aligned} \quad . \tag{4.29}$$

To complete the example, assuming $b' = 0.61$, the edge $(5,4)$ is chosen.

Although the value of b' and the probabilities have been arbitrarily assumed, it is important to highlight that the probability distribution is not uniform and is highly shifted to the most probable node. This is caused by the fact that the candidates are maximum 3, since each node has potentially 4 neighbours, but one of these neighbours is the previous node of the path. In cases where the candidates are fewer than 3, such as in boundaries or when a neighbour has already been visited, the probability of the most probable node could reach values much higher than 60%. In the classic TSP, this disuniformity of probability is not generally present since all the cities are neighbours and so the probability is more distributed.

To artificially redistribute the probability, Equation (4.6) is extended by adding another rule which favours the least probable node:

$$j = \begin{cases} \arg\max_{l \in \mathcal{N}_i^k} \{[\tau_{il}]^\alpha [\eta_{il}]^\beta\} & \text{if } b < b_0 \quad \text{(Full exploitation)} \\ J & \text{if } b_0 \leq b \leq b_1 \\ \arg\min_{l \in \mathcal{N}_i^k} \{[\tau_{il}]^\alpha [\eta_{il}]^\beta\} & \text{if } b > b_1 \quad \text{(Full exploration)} \end{cases} . \tag{4.30}$$

By means of the parameter b_1, it is possible to increase the chance of the node with lower probability, which otherwise would be highly penalised during the *roulette wheel*.

4.4.5. Paths construction

After introducing how pheromones are updated and how the next node is chosen, the last block of the algorithm is the paths construction. In the present method, the number of parallel channels (*i.e.* paths) is an input parameter and remains constant during the optimisation. In higher-order density-based methods, the parallel channels are extracted from the optimiser, thus being less constrained. However, in the present algorithm, due to the challenges presented in Section 4.4.1, this simplification is necessary to ensure that feasible paths are constructed. The automatic generation of parallel paths will be subject of investigation in the future.

For each generation, a colony of m ants sequentially searches for paths, thus at each generation m solutions are constructed and compared. When there are multiple channels, the same ant repeats the search for every channel, with the constraint of not intersecting with the other parallel paths it has already found.

In this process, the inlet node is always in the first column and the outlet node in the last column. Two different channels cannot share the same inlet or outlet node. Consequently, in a grid with 3 rows, as in Figure 4.3, it is impossible to request 4 parallel channels, since there can be a maximum of 3 distinct channel inlets and outlets.

Before explaining the path construction process, it is important to specify that when multiple channels are present, the algorithm does not always generate feasible paths that respect the constraints. Hence, path construction involves a trial and error process. If an ant does not find a feasible solution within the maximum number of trials set, that ant is ignored, and the next ant begins its search. Usually, far fewer than 50 trials are required if the number of parallel channels is not too large, though this depends on the specific case.

Before ant k begins to search, the inlet and outlet nodes of the specific path are defined. They are chosen randomly from the available nodes in the first column (inlet) and the last column (outlet). This process could alternatively be executed using the same state transition rule employed for the choice of intermediate nodes; however, doing so could cause strong reinforcement between a specific inlet-outlet pair, resulting in other possible combinations being insufficiently explored. By randomly extracting the combination of inlet and outlet, better uniformity among the different combinations is ensured, preventing possible solutions from being discarded *a priori*.

If ant k is searching for multiple paths, before extracting the inlet and outlet combination, the algorithm verifies that the nodes already visited by the ant's previously found paths leave enough available nodes in the first and last columns to generate the remaining channels. Specifically, for both the first and last columns, the number of visited nodes from the other paths (n_{vst}) must be lower equal than the difference between the total number of rows (n_{rows}) and the remaining number of channels to find (n_{chr}):

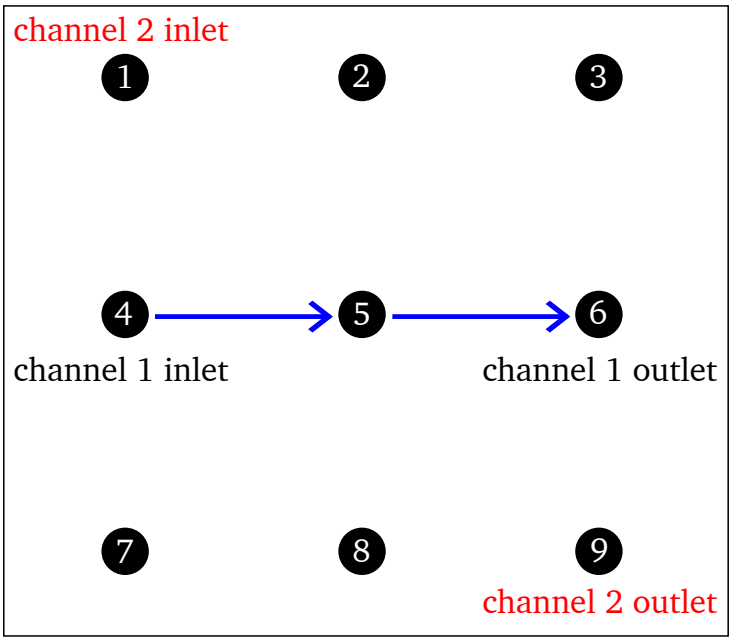


Figure 4.4: Example of a path search without a valid solution; it is impossible to connect the inlet and outlet of channel 2 (nodes 1 and 9) without intersecting the already established channel 1.

$$n_{\text{vst}} \leq n_{\text{rows}} - n_{\text{chr}} \, ; \tag{4.31}$$

if this condition is false for either the first or the last column, the search restarts from scratch with another trial.

Once the inlet and outlet of the current path are established, ant k begins its search from the inlet node using the transition rule detailed in Section 4.4.4. During the route, ant k must respect the continuity of the path by moving only to available adjacent nodes (north, south, east, west) and must not intersect with any other established paths. Figure 4.4 presents an example where ant k has already found a first path composed of nodes [4, 5, 6], and is starting the search for a second path. As mentioned, the inlet and outlet nodes are extracted randomly before the ant starts the route; in this case, for the second path, the inlet is node 1 and the outlet is node 9. The ant must find a continuous path between these two positions. However, this specific search has no valid solution, as the outlet node can only be reached by intersecting the first path, which is prohibited. Consequently, in this scenario, ant k discards all the found paths (only the first path, in this example) and initiates another trial from scratch.

Another exception involves dead ends, *i.e.*, nodes that have no available neighbours to continue the route towards the outlet node. This situation is handled by a backtracking procedure, where the ant moves backwards along its path until it finds a node that does not lead to a dead end. This procedure is depicted in Figure 4.5, which shows 4 sequential steps.

The inlet is node 4, and the target outlet is node 9. The first frame depicts a situation where the ant reaches node 1 and becomes trapped in a dead end, because from node 1, both valid neighbours (node 2 and node 4) have already been visited. In the next step, the ant retreats to the previous node (node 2), recording that node 1 leads to an unfeasible solution. From node 2, there are no remaining unvisited nodes that can lead to the outlet, so the ant retreats further to node 3. Here, the situation remains unchanged, as the only unvisited option is node 2, which has just been determined to lead to a dead end. Therefore, the ant moves back again to node 6. At node 6, the ant has two options: node 3, which leads to a dead end and is thus ignored, and node 9. Ant k therefore moves to node 9, which is also the designated outlet node.

After ant k has found a number of paths equal to the number of parallel channels specified as input (n_{ch}), there is the check on the total number of fluid nodes (n_{fn}). This check is to avoid the trivial solution where all the nodes are fluid (which leads to the minimum objective function) and to avoid solutions with short paths which have a low heat transfer area.

$$n_{\text{fn, min}} < n_{\text{fn}} < n_{\text{fn, max}} \,. \tag{4.32}$$

If the condition in Equation (4.32) is not satisfied, the ant discards the current paths and performs another trial, restarting from scratch.

4.4.6. Best solution mutation

As in most ACO applications, the proposed algorithm can remain trapped in local optima, especially when the number of nodes is large. These local optima can often be improved by minor modifications, which have a low probability of occurring due to the strong reinforcement of the best solution by the MMAS pheromone update. As discussed previously, different randomisation elements are included in the algorithm to enhance exploration; however, these elements frequently lead to worse solutions that are far from the local optima. To establish a more direct approach, a mutation of the global best solution is introduced.

The mutation occurs with a probability b_{mut}. Each time an ant begins its route, a random number b is generated. If $b < b_{\text{mut}}$, instead of searching for paths from scratch, the ant performs a mutation on the best global solution found so far. Unlike the classical mutation employed in standard genetic optimisation algorithms, which simply mixes two different discrete solutions, this approach must strictly enforce the constraints regarding path continuity and non-intersection between parallel channels. This makes the mixing of different solutions impractical; consequently, the mutation is performed exclusively on the global best path, without combining it with other solutions. When multiple parallel paths are present, the mutation is applied to only one randomly selected path. The subsequent step determines which segment of the chosen path is mutated. Using deterministic metrics would not yield significant mutations, as the altered segment would remain identical as long as

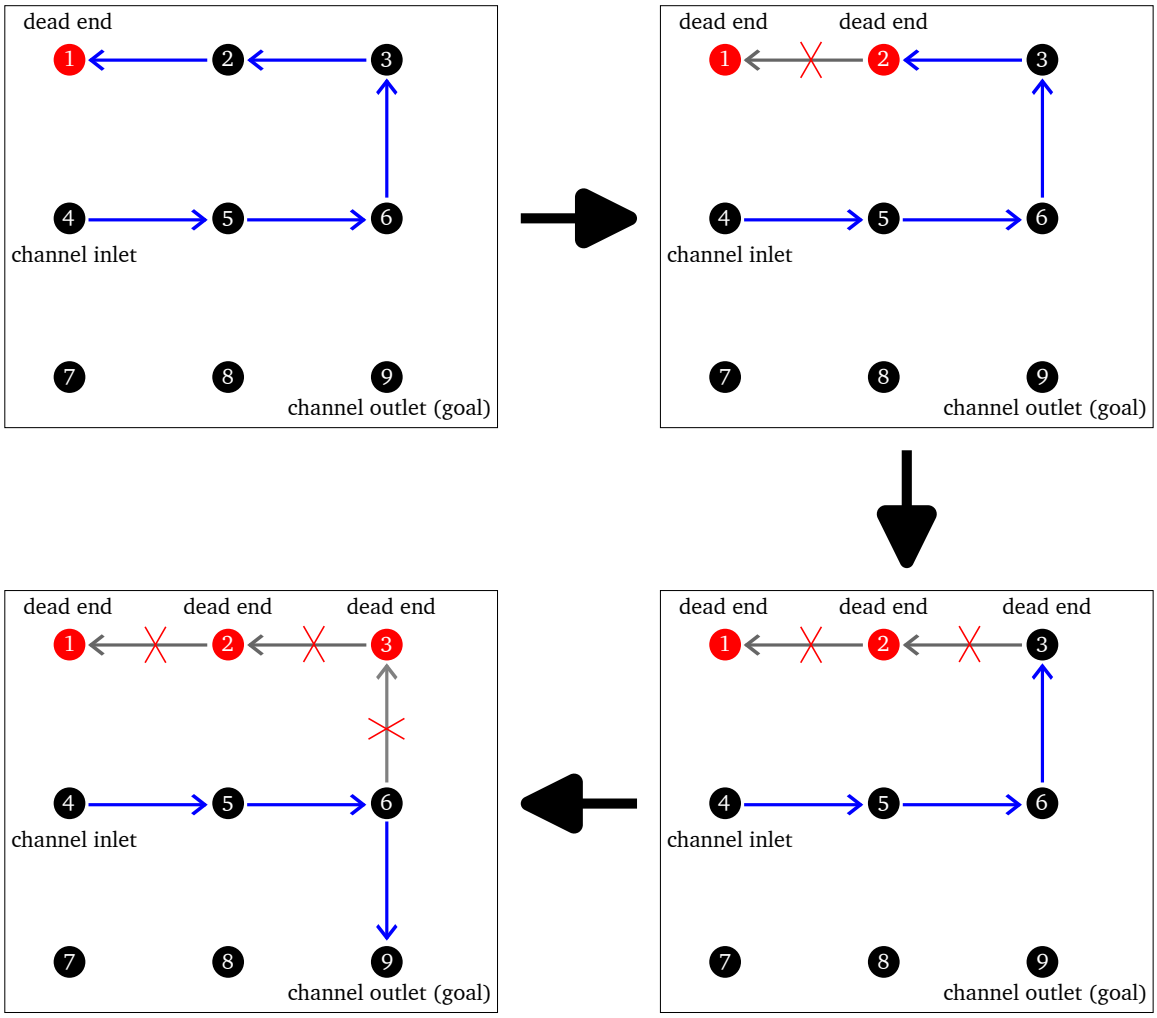


Figure 4.5: Backtracking process employed to avoid dead ends during path construction.

the global best solution remains unchanged. Therefore, a random approach is employed to select the start and end nodes for the mutation. Once these nodes are fixed, the algorithm used to construct paths can be applied. The start node for the mutation can be any node between the path inlet and two nodes prior to the path outlet, ensuring sufficient space for the mutation. If the start node is too close to the outlet node, the potential for mutation is restricted. The end node can be any node between the second node after the start node and the node immediately preceding the outlet. To guarantee a different solution, the initial edge extending from the start mutation node of the original path is discarded as a possible route.

To clarify this procedure, consider the first solution in Figure 4.6, and assume that channel 2 (nodes [1, 2, 3, 4, 8]) is randomly selected for mutation. The candidate nodes for the mutation start are [1, 2, 3]. In this example, node 1 is randomly extracted as the start node; consequently, the candidates for the end node are [3, 4]. This choice is also random, and node 4 is selected. The algorithm used to construct a path (detailed in Section 4.4.5) is then utilised to find a new path between nodes 1 and 4, maintaining the check on the

number of fluid nodes. From node 1, the ant is forced to follow a different route compared to the original path, because the edge $(1,2)$ is disabled. This mutation does not introduce large variations in the best solution; it is designed to explore the vicinity of local optima by increasing the selection probability of edges near those of the best solution. After the mutation, the ant deposits pheromones on the new edges, which subsequently attracts other ants.

Because the mutation generates solutions close to the current global best solution, it is inefficient to apply it at the beginning of the optimisation, when the exploration of widely different solutions is required. Thus, the mutation probability b_{mut} is not constant during the optimisation. It is initialised with a low value for the first generations and then increases gradually. Specifically, b_{mut} is multiplied by a factor of 1.05 each time the pheromone values are reinitialised, *i.e.*, when the global best solution does not change for n_{imp} iterations. Additionally, to prevent a large portion of the ants from focusing solely on mutation, b_{mut} is bounded by a maximum value, $b_{\text{mut,max}}$. By adjusting $b_{\text{mut,max}}$, the maximum mutation probability can be controlled, and it can be set to zero if mutation is not desired.

Figure 4.7 presents the flowchart for the paths construction subroutine. This subroutine receives as input all the parameters regarding the grid topology and number of parallel paths required. Furthermore it also receives the information whether the mutation of the best solution is active, based on the comparison between the generated random number and b_{mut}.

The main steps of the process are listed below:

i. Upon initialisation, the subroutine sets the attempt counter to zero.

ii. All paths found thus far by ant k are deleted, and the index for the number of found paths is reset to zero ($n_{\text{paths}} = 0$). Note that at the first attempt, there are no paths to delete.

iii. The attempt counter is incremented by one. If the number of attempts exceeds the maximum allowable limit, the subroutine terminates, ignoring ant k and proceeding directly to the next ant.

iv. Before starting the search, the ant requires information regarding the nodes to connect. If mutation is active, the start and end nodes are extracted randomly from a random path within the global best solution. If mutation is inactive, the inlet and outlet nodes are selected randomly from the first and last columns, ensuring they have not been visited by other paths already found by ant k. If there are insufficient candidate nodes to generate the required number of parallel paths, the search is aborted, and a new attempt begins from step 2.

v. Ant k initiates its route from the start node and continues searching until it reaches the end node. Before each move, it checks for available unvisited neighbours. If valid

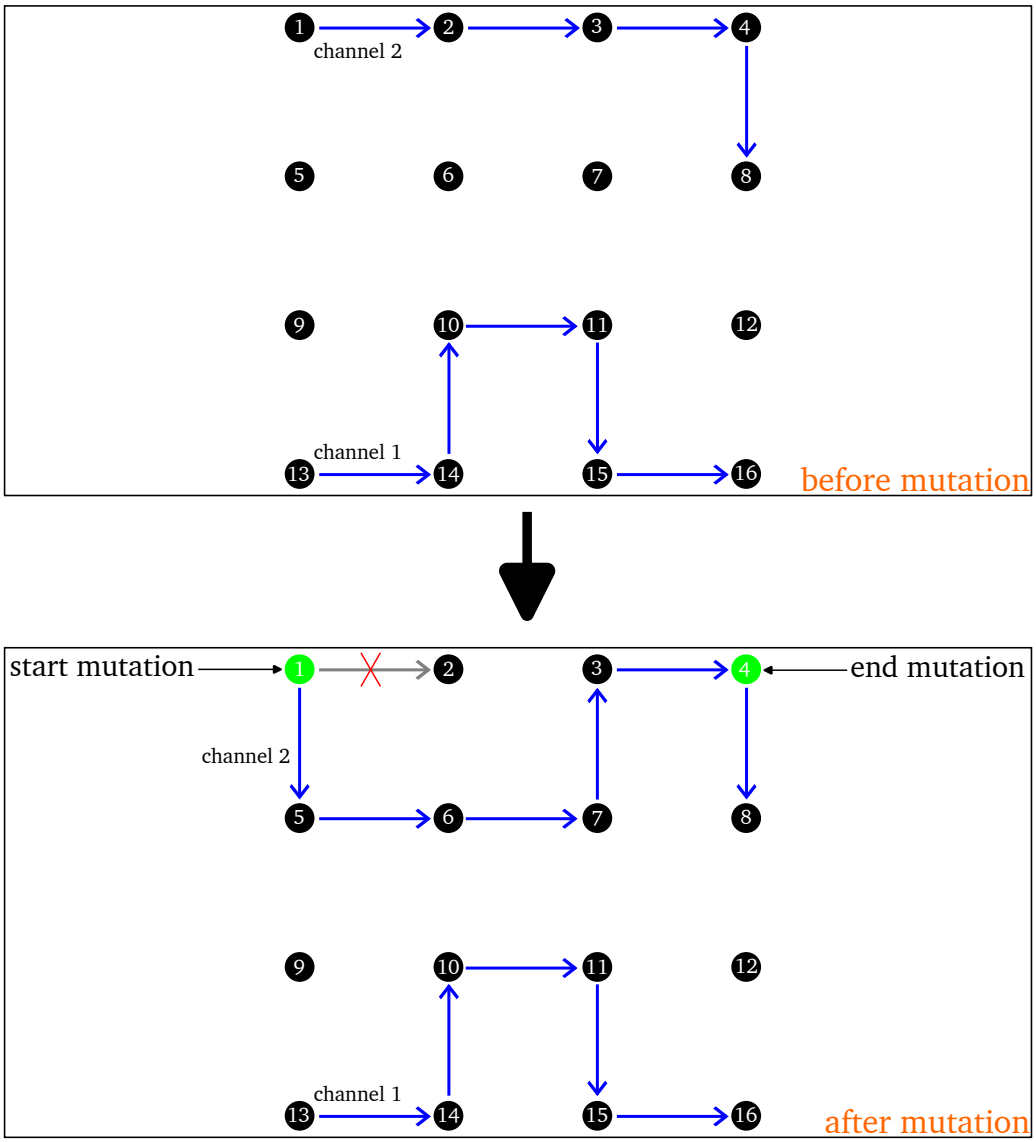


Figure 4.6: Example of mutation of a solution with 2 paths.

neighbours exist, the ant moves to one based on the state transition rule (Equation (4.30)). If no nodes are available, the ant backtracks, marking the dead end node as visited to prevent returning to it.

vi. If the ant backtracks to the start node and no other neighbours are available (or all are marked as dead ends), the current search is terminated, and a new attempt begins from step 2.

vii. If ant k reaches the end node, the total fluid volume constraint is checked ($n_{\text{fn, min}} < n_{\text{fn}} < n_{\text{fn, max}}$). If the constraint is not met, the search restarts with a new attempt from step 2. If the constraint is satisfied, two scenarios are possible: if the search involved mutating a path within the best solution, the modified best solution is returned; otherwise, the new found path is stored in a temporary list, and the counter for paths

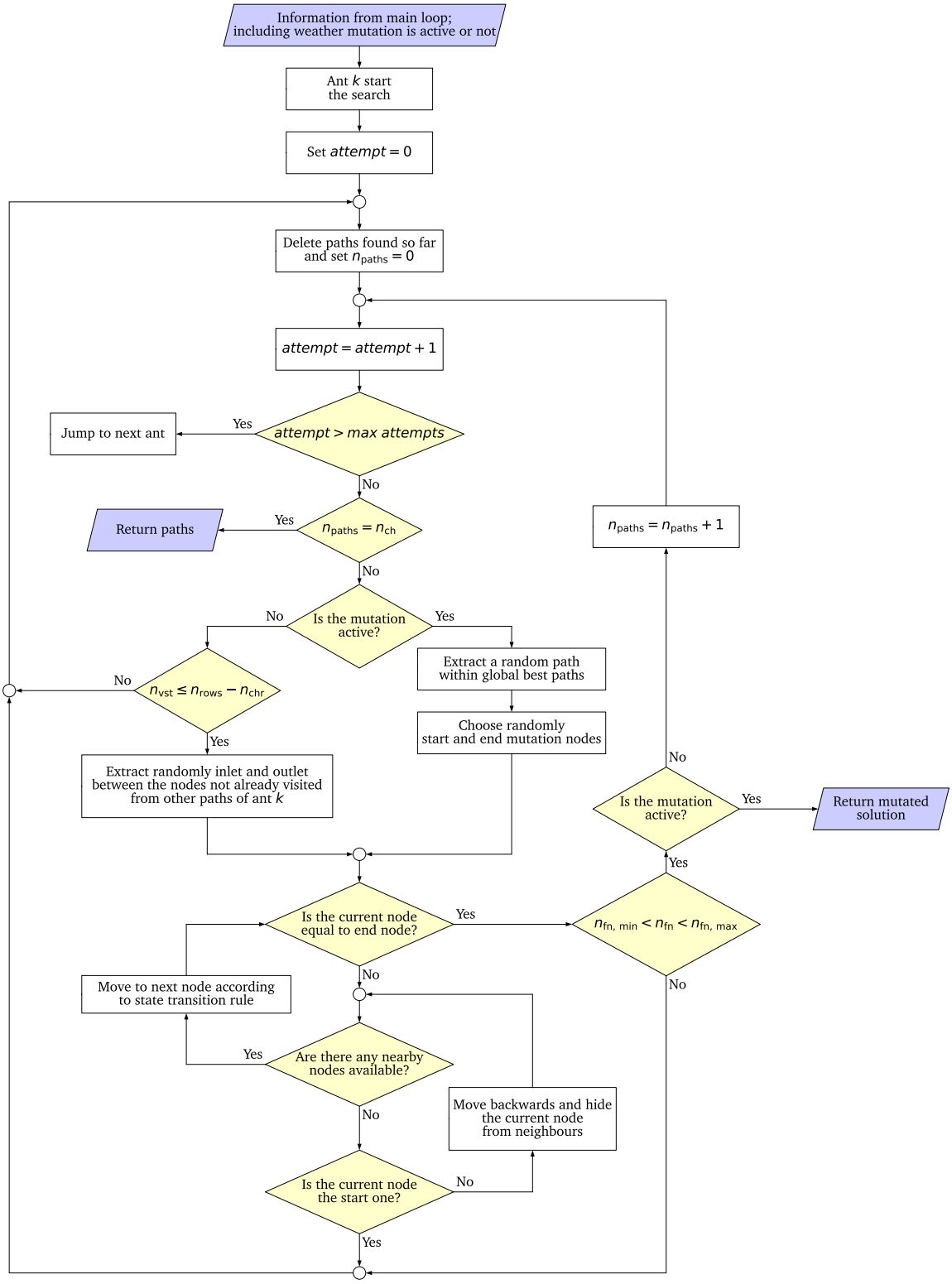


Figure 4.7: Flow chart describing the main steps of the paths finding algorithm.

found by ant k is incremented by one.

viii. If the number of paths in the temporary list equals the required number of channels (n_{paths}=n_{ch}), the subroutine terminates and returns the solution. Otherwise, the loop restarts from step 3 to find the next parallel path.

The paths finding subroutine involves a substantial number of decision processes and conditions. This complexity arises from the challenges associated with the continuous representation of cooling channels, as well as the constraints necessary to ensure physical feasibility and prevent the exploration of trivial or low-quality solutions.

4.4.7. Proposed algorithm summary

The preceding sections have detailed the various elements composing the proposed algorithm. The objective function g is defined as a weighted average of the non-dimensional average temperature difference and the normalised standard deviation of the temperature on the top plate's surface. Constraints on the maximum temperature difference and maximum head loss are defined as penalties added to the objective function, which are proportional to the magnitude of the violation. Pheromones are updated globally according to the MMAS algorithm, which updates only the iteration-best or global-best solution and bounds the maximum and minimum pheromone levels. To enhance exploration, a local pheromone update is also enabled, discouraging an ant from following the same edges as previous ants. Additionally, if the global best solution does not improve for a certain number of iterations, the pheromone levels are re-initialised, retaining a specific fraction of memory.

The state transition rule is an extension of the *pseudo-random-proportional rule*. Based on the result of a randomly generated number, this rule can return the most probable node directly, employ the *roulette wheel* method, or return the least probable node directly. This method yields a more uniform probability distribution compared to the standard *pseudo-random-proportional rule*. Finally, the path-finding algorithm is discussed; this is based on a trial-and-error procedure to find the optimal path between two randomly extracted nodes without violating constraints regarding continuity, intersections, and total fluid volume. The start and end nodes can be the inlet and outlet nodes of the first and last columns, respectively, or internal nodes of a random path from the current global best solution. Another randomly generated number determines whether the ant searches for paths from scratch or mutates the global best solution. The main steps are depicted in Figure 4.8 and can be summarised in 13 steps:

i. Pheromones are uniformly initialised to τ_0.

ii. The time (or generation index) is initialised, starting with the first generation of ants.

iii. The ant index within the generation is initialised to zero.

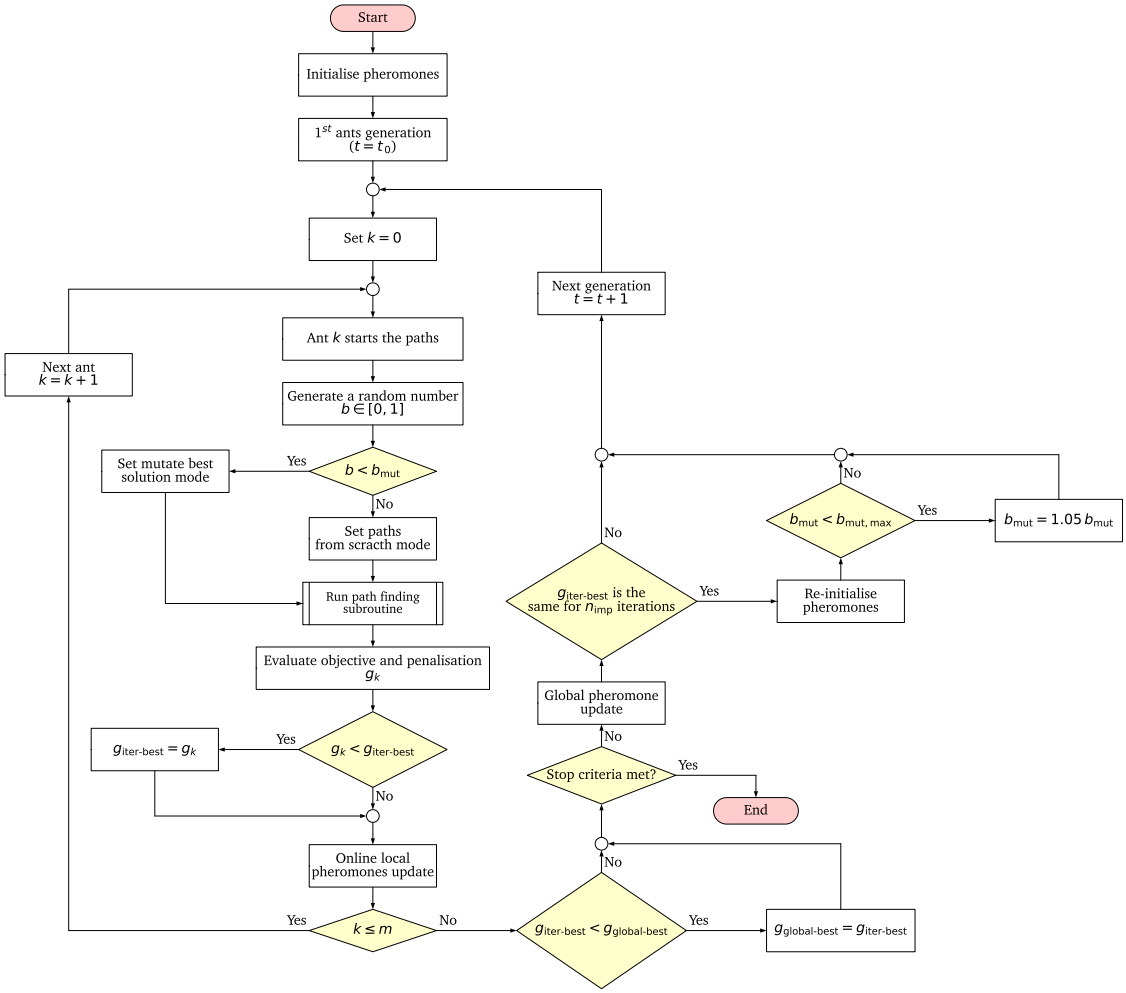


Figure 4.8: Flow chart describing the main steps of the proposed ACO.

iv. The ant begins the path search. Before starting the search, a random number b is generated; if $b < b_{\text{mut}}$, the ant performs a mutation of the current best solution. Otherwise, it searches for paths from scratch. Figure 4.7 illustrates the flowchart for the path-finding algorithm.

v. Once the ant returns a solution, the objective g_k it is evaluated by employing the equivalent thermal network. If the constraints on the maximum temperature standard deviation and head loss are violated, a penalty proportional to the magnitude of the violation is applied.

vi. If the new found solution yields an objective function value lower than the minimum objective function of the current generation, the latter is updated.

vii. Before concluding the loop for ant k, the online pheromone update (Equation (4.20)) is performed on the edges belonging to the paths just found by the ant.

viii. If not all ants in the colony have performed their search, the algorithm returns to step

4 with the next ant.

ix. Once all ants in the colony have completed the search, the best solution of the current generation is compared with the current global best solution. If the former is lower, the global best solution is updated.

x. If the global best solution remains unchanged for a specified number of iterations, or if the maximum number of iterations is reached, the optimisation terminates.

xi. If the optimisation continues, the pheromones are globally updated using Equation (4.9). At the start of the time loop, the iteration-best solution is employed. After n_{lb} iterations, a switch between the iteration-best and global-best solutions is introduced with a frequency of n_{s} iterations.

xii. If the global best solution has not improved for n_{imp} generations, the pheromone levels are re-initialised using Equation (4.21). If the mutation probability is lower than the maximum bound set, the value of b_{mut} is multiplied by a factor of 1.05.

xiii. The time index is incremented by one, and the loop restarts from step 3 with the next generation.

As for the standard ACO, the proposed algorithm does not guarantee reaching the global minimum, particularly when the solution space is large. Therefore, the standard procedure involves running multiple independent optimisations and selecting the solution that yields the minimum objective function value.

Having detailed the proposed algorithm and its solution strategy, the next section focuses on the equivalent thermal network used for solution evaluation, which utilises the same nodes grid traversed by the ants.

4.5. Network construction

Once an ant completes a tour, the new solution is evaluated by means of the thermal network solver detailed in Section 2.7. The procedure is automated using a compiled Python script, which generates and solves the network. The inputs to this script include the number of rows and columns of the starting grid, the dimensions, the physical properties of the coolant and the solid, the total flow rate, and the total heat source. Additionally, for each (rectangular) channel, an ordered list of the nodes belonging to the channel is provided along with its aspect ratio (defined as the channel width divided by the height). Since a key characteristic of cooling plates is their limited thickness, the channel height is fixed to h=5 mm, and only the width varies according to the aspect ratio.

To ensure consistency with the physics of the problem, the simulated network is not composed solely of the starting grid processed by the ACO; it also includes a top layer.

This layer represents the solid material between the channel top wall and the cooling plate top surface, where the heating is applied. For each node in the channels layer, there is a corresponding node in the top layer where a constant heat source is applied (Equation (2.163)). Figure 4.9 schematically illustrates the construction of a 3×3 grid with 2 channels. For each layer, all nodes lump the same volume, calculated as the surface area multiplied by the height of the layer:

$$\begin{aligned} \Delta x &= \frac{\text{plate width}}{n_{\text{cols}}} \\ \Delta y &= \frac{\text{plate length}}{n_{\text{rows}}} \\ V_{\text{node}} &= \Delta x \, \Delta y \, h_{\text{layer}} \, , \end{aligned} \tag{4.33}$$

where V_{node} is the volume lumped by a single node, Δx is the horizontal distance between nodes, Δy is the vertical distance, and h_{layer} is the height of the layer in which the node is located. For the channels layer, h_{layer} corresponds to the channels height (equal for all the parallel channels), while for the top layer, it is equal to the distance between the top surface of the channels and the top surface of the plate.

The use of 2 channels in this example allows for the highlighting of network features present when multiple parallel channels are present. The figure displays the channel layer with the 2 channels, and below it, a cross-sectional view showing the connection with the top solid layer.

4.5.1. Definition of thermal resistances

The network generator begins by creating the advective resistances between the fluid nodes of each channel, based on the input lists. Once all fluid nodes are connected, additional advective resistances are added to connect the inlet of each channel with the global inlet node (where the channels split), and the outlet of each channel with the global outlet node (where the channels are re-merged).

After inserting all advective resistances, the remaining resistances are automatically defined according to the nature of the two neighbouring nodes:

i. Solid-fluid connection: if one node is solid and the other is fluid (e.g., nodes 2-3), the resistance is defined as a series combination of a convective resistance (between the fluid node and the channel wall) and a solid resistance (between the channel wall and the solid node).

ii. Solid-solid connection: if both nodes are solid (e.g., nodes 1-2), they are connected via a conductive resistance.

iii. Fluid-fluid connection: when both nodes are fluid but separated by a solid layer (e.g., nodes 5-8), the connection is made by means of a series of convective-conductive-

convective resistances. This occurs when the two fluid nodes belong to different channels, or when they belong to the same channel but are not consecutive, such as in "U" shapes.

After linking all nodes in the channels layer, the top layer is created. In the top layer, all nodes are solid; therefore, all connections within this layer are made via conductive resistances. The connections to the corresponding nodes in the channels layer are conductive if the channels layer node is solid, or composed of a conductive-convective resistance if the channels layer node is fluid. The total heat source is divided uniformly among the solid nodes in the top layer.

Although the equations for the thermal resistances have already been presented in Chapters 2 and 3, they are reported here for completeness.

Advective resistances represent the temperature variations of the fluid inside the channels and are derived from the internal energy balance. Since this depends on the mass flow rate, the advective resistance varies between channels if the mass flow rate is not uniform. Therefore, the advective resistance for the i-th channel is:

$$R_{\mathrm{adv},i} = \frac{1}{\dot{m}_i \, c_{\mathrm{p}}} \,, \tag{4.34}$$

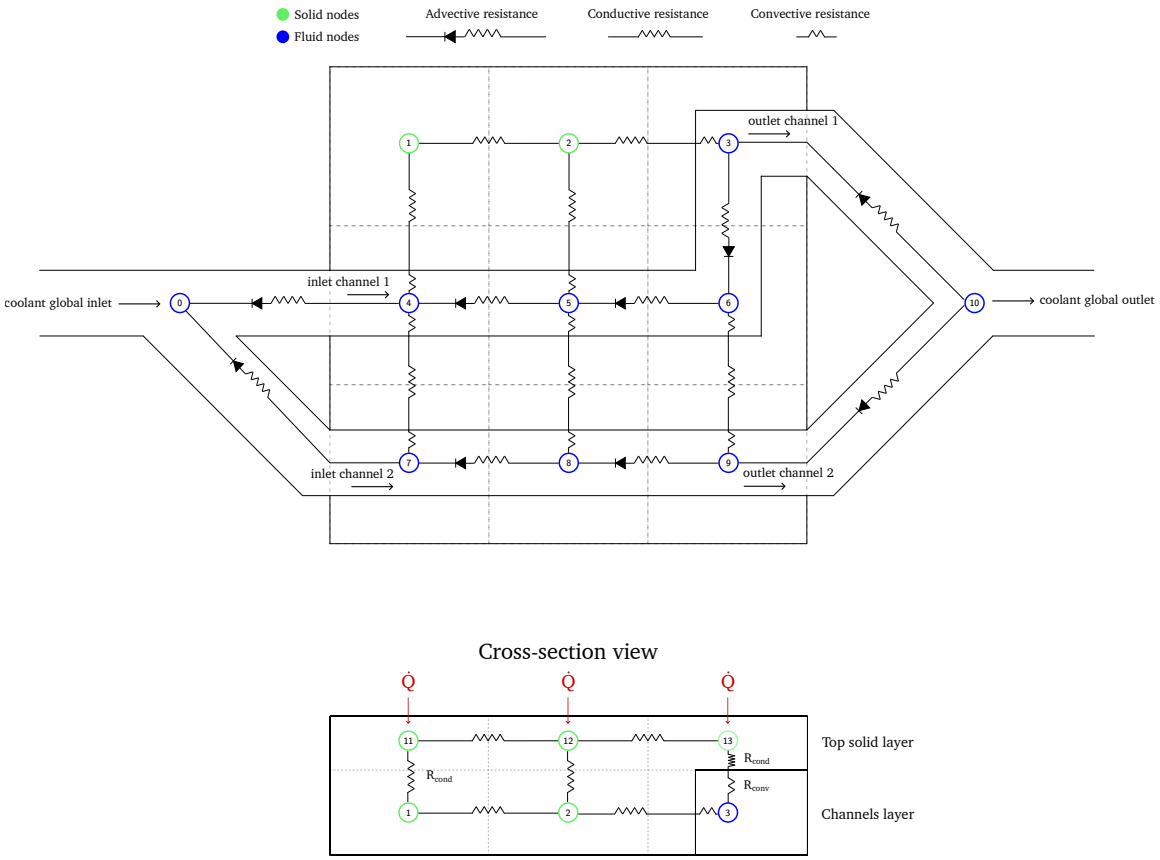


Figure 4.9: Example of a 3 × 3 network with 2 channels.

where $\dot{m}_i$ is the mass flow rate of the i-th channel and c_p is the specific heat at constant pressure of the coolant.

Conductive resistances are calculated as:

$$R_\mathrm{cond} = \frac{s}{\lambda_\mathrm{solid} A_\mathrm{cross}} , \tag{4.35}$$

where s is the material thickness, λ_solid is the thermal conductivity of the solid material, and A_cross is the cross-sectional area of the material layer.

The convective heat transfer resistance is calculated as:

$$R_{\mathrm{conv},i} = \frac{1}{U_i A_\mathrm{conv}} , \tag{4.36}$$

where A_conv is the convective heat transfer area and U_i is the convective heat transfer coefficient for the i-th channel:

$$U = \frac{\mathrm{Nu}\,\lambda_\mathrm{fluid}}{D_{\mathrm{h},i}} . \tag{4.37}$$

The hydraulic diameter $D_{\mathrm{h},i}$ of the i-th channel is:

$$D_{\mathrm{h},i} = \frac{2\,w_i\,h_i}{w_i + h_i} , \tag{4.38}$$

with w and h referring to the channel width and height, respectively.

The Nusselt number Nu is calculated by means of empirical correlations. If the Reynolds number inside the specific channel is $\mathrm{Re}_i < 3 \times 10^3$, a constant value is imposed: Nu_i=4.36 [220]. Conversely, when $\mathrm{Re}_i \geq 3 \times 10^3$, the relation proposed by Gnielinski [221] is employed.

$$\mathrm{Re}_i = \frac{\bar{u}_i\,D_{\mathrm{h},i}}{\nu} , \tag{4.39}$$

with $\bar{u}_i$ being the mean stream-wise fluid velocity in the i-th channel and ν the coolant kinematic viscosity.

$$\mathrm{Nu}_i = \begin{cases} 4.36 & \text{if } \mathrm{Re}_i < 3 \times 10^3 \\ \dfrac{(f_i/8)(\mathrm{Re}_i - 1000)\,\mathrm{Pr}}{1 + 12.7\,(f_i/8)^{0.5}\,(\mathrm{Pr}^{2/3} - 1)} & \text{if } 3 \times 10^3 \leq \mathrm{Re}_i \leq 5 \times 10^6\,,\ 0.5 \leq \mathrm{Pr} \leq 2 \times 10^3 \end{cases} , \tag{4.40}$$

where Pr denotes the Prandtl number of the coolant and f_i represents the friction factor inside the i-th channel. The latter is determined via Equation (4.42), proposed by Churchill [222] to approximate the implicit Colebrook-White equation [134]. Although the Churchill correlation is slightly less accurate with respect to the one proposed by Colebrook, it is employed in the present algorithm since it is valid for all flow regimes and therefore it ensure a smooth transition of the friction factor value between laminar and turbulent regime. This

feature is very useful, since it avoids numerical instabilities in the implicit method governing the flow rate splitting (detailed in Section 4.5.2). Conversely, using a step function defined by the laminar equation (f=64/Re) for laminar flows, and by the Colebrook equation (Equation (2.101)) for turbulent flows, could cause convergence failure when transitional regime occur.

$$\mathrm{Pr} = \frac{\nu \rho \, c_\mathrm{p}}{\lambda_\mathrm{fluid}} \, . \tag{4.41}$$

$$\begin{cases} f_i = 8\left[\left(\dfrac{8}{\mathrm{Re}_i}\right)^{12} + \dfrac{1}{(A+B)^{1.5}}\right]^{1/12} \\ A = \left[-2.457 \ln\left(\left(\dfrac{7}{\mathrm{Re}_i}\right)^{0.9} + 0.27\dfrac{Ra_i}{D_{\mathrm{h},i}}\right)\right]^{16} \\ B = \left(\dfrac{37530}{\mathrm{Re_i}}\right)^{16} \end{cases} , \tag{4.42}$$

In the above equations, Ra_i represents the surface roughness of the channel and ρ is the coolant density.

4.5.2. Flow rate splitting

When multiple parallel channels are present, the flow rate distribution among them is initially unknown. An exception occurs only when all channels share identical geometric properties (length, number of turns) and roughness; in such cases, the total flow rate is divided equally among the channels. However, when these parameters differ, the flow rate distribution is non-uniform. Under steady-state conditions, channels with lower hydraulic resistance present a higher flow rate to satisfy the condition of equal pressure drop across the shared inlet and outlet sections.

For $i \in [1, n_\mathrm{ch}]$, where n_ch is the number of parallel channels, the following conservation rules apply:

$$\begin{aligned} \dot{V}_\mathrm{tot} &= \dot{V}_1 + \dot{V}_2 + \ldots + \dot{V}_i + \ldots + \dot{V}_{n_\mathrm{ch}} \\ \Delta p_1 &= \Delta p_2 = \ldots = \Delta p_i = \ldots = \Delta p_{n_\mathrm{ch}} \end{aligned} , \tag{4.43}$$

where $\dot{V}$ denotes the volumetric flow rate and Δp represents the pressure drop between the inlet and outlet sections. The pressure drop in each channel is calculated as the sum of distributed losses and concentrated losses induced by the turns:

$$\Delta p_i = \Delta p_{\mathrm{d},i} + \Delta p_{\mathrm{c},i} = f_i \frac{L_i}{D_{\mathrm{h},i}} \rho \frac{\bar{u}_i^2}{2} + n_{\mathrm{t},i} K \rho \frac{\bar{u}_i^2}{2} \, , \tag{4.44}$$

where L_i is the total length of the i-th channel, $n_{\mathrm{t},i}$ is the number of turns in the i-th channel, and K is the concentrated loss coefficient. In this study, concentrated losses

are assumed to arise solely from turns; therefore, K is treated as constant and is equal to $K = 2.0$ [171], which corresponds to the typical value for 90° curves. Thus, the total concentrated loss for each channel is evaluated by multiplying the loss coefficient of a single turn by the number of turns in that channel. By utilising the relations in Equation (4.43), it is possible to write:

$$\begin{cases} \Delta p_1 = \Delta p_2 \\ \Delta p_1 = \Delta p_3 \\ \vdots \\ \Delta p_1 = \Delta p_{n_{\mathrm{ch}}} \end{cases} . \tag{4.45}$$

This system comprises $n_{\mathrm{ch}} - 1$ equations for n_{ch} unknowns, where the unknowns are the flow rates in each channel appearing in the pressure drop equations. It is recalled that the mean stream-wise velocity and the volumetric flow rate are related by:

$$\bar{u}_i = \frac{\dot{V}_i}{A_i} , \tag{4.46}$$

where A_i is the internal cross-sectional area of the channel. To close the system, the flow rate conservation equation is employed, yielding:

$$\begin{cases} \dot{V}_{\mathrm{tot}} = \dot{V}_1 + \dot{V}_2 + \ldots + \dot{V}_i + \ldots + \dot{V}_{n_{\mathrm{ch}}} \\ \Delta p_1 = \Delta p_2 \\ \Delta p_1 = \Delta p_3 \\ \vdots \\ \Delta p_1 = \Delta p_{n_{\mathrm{ch}}} \end{cases} . \tag{4.47}$$

The system now consists of n_{ch} equations and n_{ch} unknowns. To explicitly express the dependence of Equation (4.45) on velocity, the coefficient C is introduced:

$$C_i = f_i \frac{L_i}{D_{\mathrm{h},i}} \frac{\rho}{2} + n_{\mathrm{t},i} K \frac{\rho}{2} . \tag{4.48}$$

Consequently, the equations in (4.45) take the form:

$$C_1 \bar{u}_1^2 = C_i \bar{u}_i^2 . \tag{4.49}$$

To linearise the dependence on velocity, the square root of Equation (4.49) is taken, yielding:

$$\sqrt{C_1}\,|\bar{u}_1| = \sqrt{C_i}\,|\bar{u}_i| . \tag{4.50}$$

Since the analysis concerns the stream-wise mean velocity, negative values of $\bar{u}$ (representing reverse flow) are neglected, implying $|\bar{u}_i|=\bar{u}_i$. Substituting $\bar{u}$ using Equation (4.46) yields:

$$\sqrt{C_1}\,\frac{\dot{V}_1}{A_1} = \sqrt{C_i}\,\frac{\dot{V}_i}{A_i}\,. \tag{4.51}$$

The system of equations (4.47) then becomes:

$$\begin{cases} \dot{V}_1 + \dot{V}_2 + \ldots + \dot{V}_i + \ldots + \dot{V}_{n_{\text{ch}}} = \dot{V}_{\text{tot}} \\ \dfrac{\sqrt{C_1}}{A_1}\,\dot{V}_1 - \dfrac{\sqrt{C_2}}{A_2}\,\dot{V}_2 = 0 \\ \dfrac{\sqrt{C_1}}{A_1}\,\dot{V}_1 - \dfrac{\sqrt{C_2}}{A_3}\,\dot{V}_3 = 0 \\ \vdots \\ \dfrac{\sqrt{C_1}}{A_1}\,\dot{V}_1 - \dfrac{\sqrt{C_{n_{\text{ch}}}}}{A_{n_{\text{ch}}}}\,\dot{V}_{n_{\text{ch}}} = 0 \end{cases}. \tag{4.52}$$

In matrix notation, the system in Equation (4.52) is expressed as:

$$\boldsymbol{C}\,\dot{\boldsymbol{V}} = \boldsymbol{r}\,, \tag{4.53}$$

where the coefficient matrix is an $n_{\text{ch}} \times n_{\text{ch}}$ matrix of the form:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ \frac{\sqrt{C_1}}{A_1} & -\frac{\sqrt{C_2}}{A_2} & 0 & \ldots & 0 \\ \frac{\sqrt{C_1}}{A_1} & 0 & -\frac{\sqrt{C_3}}{A_3} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\sqrt{C_1}}{A_1} & 0 & 0 & \ldots & -\frac{\sqrt{C_{n_{\text{ch}}}}}{A_{n_{\text{ch}}}} \end{bmatrix}. \tag{4.54}$$

The vector of unknowns contains all the channel flow rates:

$$\dot{\boldsymbol{V}} = \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \vdots \\ \dot{V}_{n_{\text{ch}}} \end{bmatrix}, \tag{4.55}$$

while the known terms vector contains only one non-zero element at the first position:

$$\boldsymbol{r} = \begin{bmatrix} \dot{V}_{\text{tot}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.56}$$

The system in Equation (4.53) is implicit, since the elements on the coefficient matrix depends on the flow rate (required to calculate the friction factor), which is also the unknown. For this reason the system is solved by means of an iterative procedure. Once the volumetric flow rate (and consequently the velocity) for each channel is obtained, the pressure drop across a single channel is calculated using Equation (4.44). This value corresponds to the total pressure drop between the global inlet and the global outlet.

4.6. Sensitivity analysis

ACO-based algorithms are simple, robust, and effective; however, they rely on a large number of parameters for which no general tuning rules exist. Consequently, the influence of these parameters requires investigation for each specific problem type. Given the absence of similar ACO applications in the literature, the optimal parameters for the optimisation performed in this chapter must be assessed. The following section discusses the sensitivity analysis regarding the most critical parameters.

4.6.1. Sensitivity on the number of nodes and channels

To determine the optimal grid density and number of channels, a sensitivity analysis is conducted comparing five different grid sizes: 5×5, 8×8, 10×10, 12×12, and 15×15. The 5×5 grid represents the minimum viable size, as smaller grids yield trivial paths that do not justify the use of this algorithm, particularly with multiple parallel channels. The maximum grid size of 15×15 was selected as the upper limit because using a higher number of nodes incurs excessive computational costs without offering benefits for a reduced-order method. Increasing the grid size causes the search space to grow exponentially, requiring significantly more generations to reach convergence. For each grid size, results were compared for 1, 2, and 3 parallel channels. The maximum number of channels was limited to 3, as using more channels tends to degrade solution quality; in such cases, the algorithm becomes more constrained by the non-intersection requirement than by the optimisation of the objective function. The parameters regarding plate geometry, material properties, and flow conditions remain consistent with the cold plate used to cool the battery packs analysed in the previous chapter (Table 3.2).

For each configuration, 20 independent runs were performed to collect statistics on

the minimum, average, and standard deviation of the objective function values. Table 4.2 lists the input parameters for the proposed algorithm. These values are selected following preliminary testing.

As defined in Equation (4.12), the objective function is a weighted average of the normalised temperature difference and the normalised temperature standard deviation. In the present application, equal weights (0.5) are assigned to both terms, reflecting the importance of temperature uniformity for battery packs efficiency and the prevention of rapid degradation. The constraints are set to a maximum temperature standard deviation of 5.0 °C and a maximum head loss of 1.0 bar. Additionally, the number of fluid nodes is constrained to be between 50% and 80% of the total nodes within the channels layer.

To scale the search effort with the solution space, the number of ants per generation is set equal to the product of the number of columns and rows. The deposited pheromone gain is set to ensure values greater than 1, avoiding numerical issues when raising values to the power of α and β. The same coefficient is used to penalise violations of both the standard deviation and head loss constraints. The initial pheromone level is uniform and set to a high value ($\tau_0 = 1 \times 10^4$) to encourage extensive exploration during the first generation. Subsequently, pheromones are updated and bounded, meaning τ_0 influences only the initial generation of ants.

As described in Equation (4.30), the movement of an ant depends on a random variable b. If $b < b_0$, the ant selects the neighbour with the highest probability; if $b > b_1$, it selects the least probable neighbour. Intermediate values leads to the *roulette wheel* selection method. Given the high number of elements introduced to enhance exploration, the probability of selecting the least probable candidate is set to 1.0% ($b_1 = 0.99$). Similarly, to prevent excessive exploitation, the probability of selecting the most probable node is limited to 10% ($b_0 = 0.10$).

To prioritise exploration at the start of the optimisation, the initial mutation probability is set to a low value ($b_{\text{mut}} = 0.05$), preventing the waste of ants on mutating suboptimal initial solutions. At each re-initialisation, this probability is multiplied by a factor of 1.05 until it reaches a maximum of 30% ($b_{\text{mut,max}} = 0.30$).

Finally, the optimisation terminates if the global best solution does not improve for 6 000 generations. While this threshold is higher than in typical ACO applications found in the literature, the expansive solution space associated with high-order grids necessitates a prolonged exploration phase to avoid premature convergence.

Parameters not discussed here, such as exponents and the evaporation coefficient, are discussed in the subsequent section (Section 4.6.2).

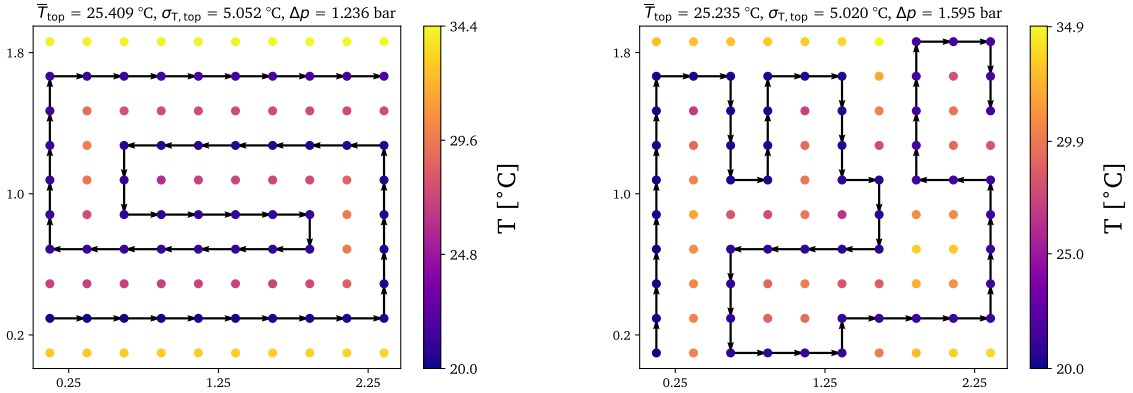
Table 4.3 shows the results of the sensitivity analysis on the number of nodes and parallel channels. In the table, the minimum value of the objective function, the average, and the standard deviation over 20 runs are listed for each case. For each metric, the table presents

Table 4.2: Parameters used for the sensitivity analysis on grid order and number of parallel channels.

Parameter	Value
Number of independent runs	20 runs
Reference temperature std. dev.	$\sigma_{\mathrm{T,ref}} = 5.0\,^{\circ}\mathrm{C}$
Inlet temperature	$T_{\mathrm{in}} = 20.0\,^{\circ}\mathrm{C}$
Temperature difference weight	$\omega_1 = 0.5$
Std. dev. weight	$\omega_2 = 1 - \omega_1$
Max. std. dev.	$\sigma_{\mathrm{T,max}} = 5.0\,^{\circ}\mathrm{C}$
Max. head losses	$\Delta p_{\mathrm{max}} = 1.0 \times 10^5\,\mathrm{Pa}$
Min. fluid nodes n.	$n_{\mathrm{fn,min}} = (0.5\, n_{\mathrm{rows}}\, n_{\mathrm{cols}})$ nodes
Max. fluid nodes n.	$n_{\mathrm{fn,max}} = (0.8\, n_{\mathrm{rows}}\, n_{\mathrm{cols}})$ nodes
Total heat source	$\dot{Q}_{\mathrm{tot}} = 2.0 \times 10^3\,\mathrm{W}$
Number of ants	$m = (n_{\mathrm{rows}}\, n_{\mathrm{cols}})$ ants
Pheromones exponent	$\alpha = 1.0$
Heuristic function exponent	$\beta = 1.0$
Min. heuristic value	$\eta_{\mathrm{min}} = 0.2$
Deposited pheromones gain	$Q = 1.0 \times 10^3$
Pressure penalisation coeff.	$Q_{\mathrm{p},\Delta\mathrm{p}} = 1.0 \times 10^3$
Std. dev. penalisation coeff.	$Q_{\mathrm{p},\sigma_{\mathrm{T}}} = 1.0 \times 10^3$
Initial pheromones level	$\tau_0 = 1.0 \times 10^4$
Evaporation coefficient	$\varphi = 0.01$
Max-min pheromones ratio	$\tau_{\mathrm{ratio}} = 20$
No improvement iter. (restart)	$n_{\mathrm{imp}} = 30$ iters
Restart memory factor	$b_{\mathrm{res}} = 0.9$
Full exploitation probability	$b_0 = 0.1$
Full exploration probability	$b_1 = 0.99$
Initial mutation probability	$b_{\mathrm{mut}} = 0.05$
Max. mutation probability	$b_{\mathrm{mut,max}} = 0.3$
Iteration with only iter-best	$n_{\mathrm{lb}} = 2000$ iters
Switch frequency iter/global-best	$n_{\mathrm{s}} = 50$ iters
Convergence criteria	6 000 iters without improvement

the actual value processed by the algorithm (including penalisation) and the re-calculated objective without penalisation. This comparison is necessary to illustrate the true magnitude of the standard deviation between runs; with penalisation, these values appear artificially high because penalties are multiplied by a factor of 1×10^3.

As shown in Figure 4.10, the layout of the best solution can change when penalisation is not taken into account. Figure 4.10a shows the best layout considering penalisation for a 10×10 grid with a single channel. The solution features a serpentine path concentrated in



(a) Best solution with penalisation.

(b) Best solution without penalisation.

Figure 4.10: Results of the algorithm with a 10 × 10 grid and a single channel; the images show the best result considering the penalisation on temperature standard deviation and head losses, and the best solution without penalisation.

the centre of the plate, ensuring good temperature uniformity in that region. The solution in Figure 4.10b represents the best result when the objective function is re-calculated *a posteriori* without considering penalisation. This solution features a more irregular path compared to Figure 4.10a, with more frequent changes in direction. In both solutions, the ants attempted to maximise the covered surface area by distributing fluid nodes uniformly. Consequently, the two solutions lead to similar results in terms of the average value and standard deviation of the temperature on the plate's top surface; however, the solution in Figure 4.10b, due to the more frequent direction changes, leads to greater head losses and is therefore more heavily penalised. Regarding the constraint on maximum temperature difference, both solutions exhibit minimal violations: 0.052 °C for the first solution and 0.020 °C for the second one. This highlights that directly discarding solutions which do not strictly respect the constraints could lead to the loss of optimal solutions that have acceptable low violations.

From Table 4.3, it can be noted that with the 5 × 5 grid and 1 channel, all 20 runs converged to the same solution; in fact, the standard deviation is zero. Figure 4.11 depicts this solution, where, as in the 10 × 10 case, the ants try to maximise surface coverage with a uniform distribution of fluid nodes. Since the plate surface is discretised into fewer nodes and therefore is larger the distance between points, the temperature standard deviation is 2.7 °C higher than the constraint limit, leading to a penalisation of 543.76 units. The pressure constraint, has a low violation as the low number of nodes does not permit a high number of curves, leading to a penalisation of 33.0 units. Regarding the average temperature, this solution gives a result similar to that obtained with the 10 × 10 grid. Thus, from this initial comparison, it seems that the number of grid points does not significantly affect the average

Table 4.3: Results of the sensitivity test on the number of nodes and channels; for each case, 20 runs were performed.

	With penalties			Without penalties		
Grid and channels	g_{min}	g_{avg}	g_{dev}	g_{min}	g_{avg}	g_{dev}
5 × 5 – 1 ch.	578.883	578.883	0.0	2.277	2.277	0.0
5 × 5 – 2 ch.	969.223	995.444	57.933	3.009	3.035	0.055
5 × 5 – 3 ch.	1949.730	1953.431	5.654	4.153	4.158	0.00801
8 × 8 – 1 ch.	410.67	522.829	67.821	1.631	1.768	0.0524
8 × 8 – 2 ch.	1.132	19.375	69.241	1.323	1.360	0.163
8 × 8 – 3 ch.	198.031	284.246	48.706	1.787	1.920	0.0738
10 × 10 – 1 ch.	248.463	566.580	112.024	2.0	2.077	0.0511
10 × 10 – 2 ch.	0.922	0.272	0.0298	0.922	0.272	0.0298
10 × 10 – 3 ch.	1.240	1.305	0.0615	1.240	1.305	0.0615
12 × 12 – 1 ch.	530.740	645.205	67.213	1.772	1.898	0.0587
12 × 12 – 2 ch.	0.644	0.757	0.040	0.644	0.757	0.040
12 × 12 – 3 ch.	0.941	1.013	0.058	0.941	1.013	0.058
15 × 15 – 1 ch.	847.010	1045.499	110.306	1.332	1.649	0.192
15 × 15 – 2 ch.	0.567	0.579	0.00564	0.567	0.579	0.00564
15 × 15 – 3 ch.	0.700	0.730	0.0140	0.700	0.730	0.0140

temperature. However, a higher node count allows for a better distribution of fluid nodes, potentially leading to lower temperature differences, albeit with higher head losses.

The behaviour differs when multiple parallel channels are present; in this case, the average temperature is also affected by the grid order, as shown in Figure 4.12. Due to the low number of nodes in the 5 × 5 grid, the ants cannot achieve temperatures similar to the 10 × 10 grid because they must avoid path intersections, preventing them from ensuring the same coverage as in the single-channel case. Regarding the 10 × 10 grid, the solution improves significantly with more parallel channels due to a better distribution of fluid nodes. The parallel configuration offers lower head losses, permitting more directional changes. In this case, both the temperature standard deviation and head losses are safely below the maximum values. The case with 2 channels appears to be the best trade-off between the number of parallel channels and number of nodes, allowing free exploration by the ants without a high risk of intersection between parallel paths.

Looking at the results in Table 4.3, using two parallel channels leads to better results for the 8 × 8, 12 × 12, and 15 × 15 grids as well. In addition the overall results improve when more nodes are used. Comparing Figures 4.12b and 4.12d with Figures 4.13a and 4.13b, it can be noted that the 12 × 12 grid yields an improved combined effect. The higher number of nodes permits the development of more features, offering a higher heat transfer area between solid and fluid and decreasing the number of hot-spots. Although the higher heat transfer area (which depends on the total path length) leads to higher pressure losses; however, the pressure difference remains at a low value.

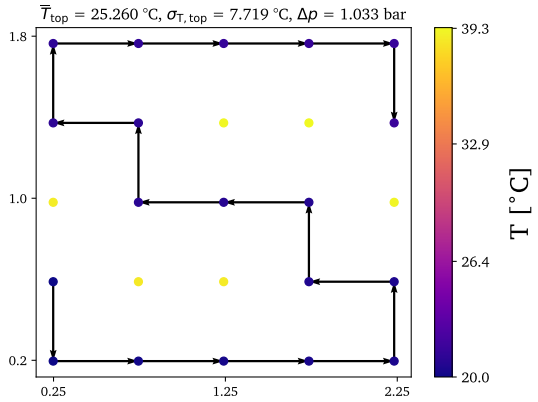
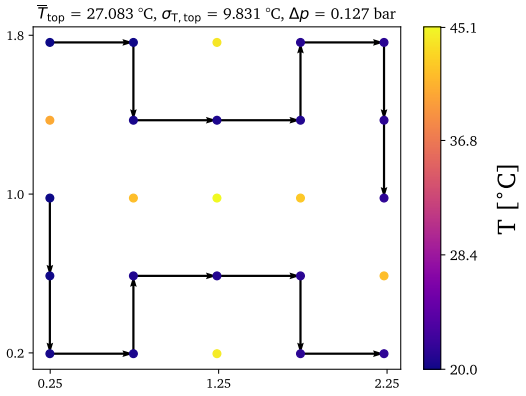
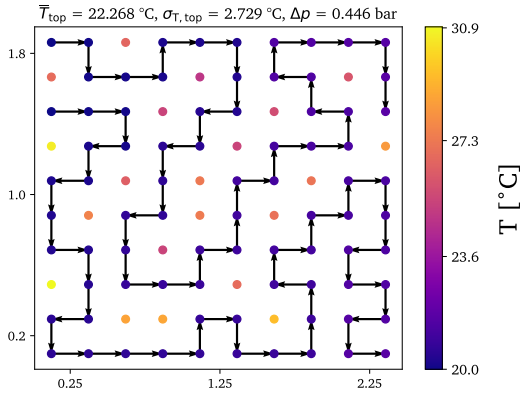


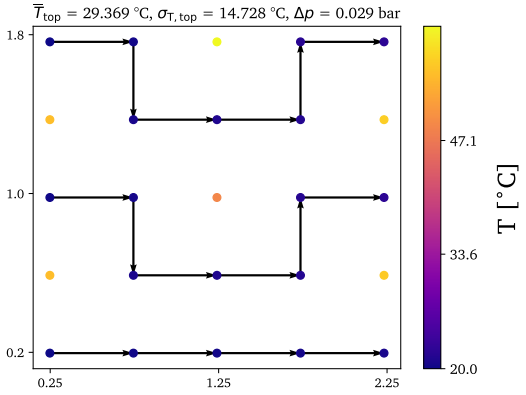
Figure 4.11: Best solution of the 5 × 5 grid with 1 channel.



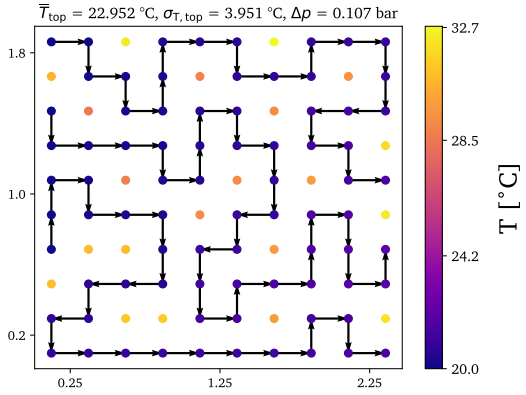
(a) Best solution with the 5 × 5 grid and 2 channels.



(b) Best solution with the 10 × 10 grid and 2 channels.



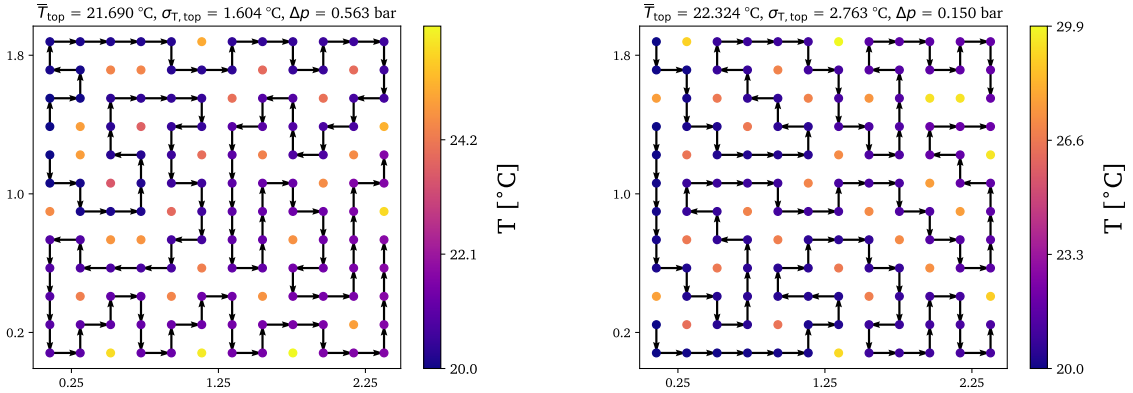
(c) Best solution with the 5 × 5 grid and 3 channels.



(d) Best solution with the 10 × 10 grid and 3 channels.

Figure 4.12: Comparison between the best solutions obtained employing grid of 5 × 5 and 10 × 10 nodes with 2 and 3 parallel channels.

Apart from the case with the 5 × 5 grid and 1 channel, in all other cases, the standard deviation of the objective function within the 20 runs is non-zero. In the case of the 5 × 5



(a) Best solution with 2 channels.

(b) Best solution with 3 channels.

Figure 4.13: Results of the algorithm with a 12×12 grid featuring 2 and 3 parallel channels.

grid solutions, is expected that the standard deviation is higher with 1 channel, as, due to the constraints on non-intersection, the solution space is effectively smaller when more channels are involved. This unexpected behaviour is due to the fact that when multiple parallel channels are present, ants are more easily trapped in local minima. The intersection constraint reduces the algorithm's capacity to explore solutions with topologies significantly different from a local minimum that already respects the constraints. However, in cases with a higher number of nodes, this behaviour is not as pronounced as in the 5×5 grid; the results show that the standard deviation averages between 1–5% for almost all the cases, apart the configurations with the 8×8 grid with 2 channels and 15×15 with 1 channel. From the sensitivity results, the configuration which leads to the lower objective function is the one featuring a 15×15 grid with 2 channels; this configuration will be used to carry out the optimisation.

4.6.2. Sensitivity on the main parameters

This section addresses the sensitivity analysis of several algorithm parameters: the (α, β) couple, φ, τ_{ratio}, n_{imp}, b_{res}, and n_s. Table 4.4 lists the analysed parameters and the specific values tested. Similar to the previous tests on grid size and channel count, the optimisation is repeated 20 times for each parameter value.

Following the methodology in [212], these parameters are tested under the *ceteris paribus* assumption: for each test, only the parameter in question is varied while all others remain fixed; this ensures that any observed variations in algorithm performance can be directly attributed to the specific parameter being analysed. Given the high number of simulations required (20 runs per value), this sensitivity analysis is conducted using a 10×10 grid with two parallel channels. This configuration represents the optimal trade-off between grid density, channels number, and computational time for 20 runs per parameter value.

Table 4.4: Values tested in the sensitivity analysis on the algorithm parameters.

Parameter	Tested values
(α, β) couple	All combination between $\alpha \in [0,5]$ and $\beta \in [0,5]$
Evaporation coeff. φ	0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7
Max-min pheromones ratio τ_{ratio}	5.0, 10.0, 20.0, 40.0, 70.0, 100.0
No improvement iter. n_{imp}	10, 30, 50, 70, 90
Restart memory factor b_{res}	0.0, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0
Switch frequency iter/global-best n_{s}	0, 10, 30, 50, 80, 100

Figure 4.14 shows the results of the runs with different (α, β) couples. The upper plot shows the average value in 20 runs, while the lower plot shows the minimum value. The dependence of the solution on the (α, β) values does not show a clear trend. On average, the lowest objective function values are obtained with $\alpha = 1$, $\alpha = 2$, and $\alpha = 5$. Conversely, $\alpha = 0$ yields the worst solutions because the algorithm ignores pheromone information and relies solely on heuristic search. However, this is not entirely consistent when observing the minimum values. In that case, $\alpha = 3$ leads to higher minimum values compared to $\alpha = 0$. This outcome can be attributed to the high degree of exploration in the algorithm and the extensive number of iterations required to reach convergence (6 000 iterations), which decreases the influence of these parameters. The lowest values in the bottom plot correspond to $\alpha = 2$, $\alpha = 4$, and $\alpha = 5$. Regarding the value of β, the results with $\beta = 0$ are superior to others only when not combined with $\alpha = 3$. On average, the values of β that yield the best results are $\beta = 0$, $\beta = 1$, and $\beta = 5$. Among the minimum scores obtained, $\beta = 4$ consistently corresponds to the lower values, along with $\beta = 3$.

Examining the combined effect of α and β, it is difficult to find similarities between the average and minimum values. The only consistent couple in both plots is (5,4), which ranks among the lowest values for both the average and the minimum. Consequently, the optimisation carried out in this chapter will use the couple (5,4) as well as the couple (1,1), which provides the lowest average score. However, the analysis suggests that higher values of α and β should be tested to investigate if results improve.

Figure 4.15 plots the results of the sensitivity analysis for the remaining parameters. For each parameter, the histogram displays the average score obtained across 20 runs, along with indications of the maximum and minimum values. Figure 4.15a indicates that the evaporation coefficient $\varphi = 0.01$ yields both the lowest average and lowest minimum values. The value $\varphi = 0.05$ leads to a slightly higher objective function. From $\varphi = 0.1$ onwards, the average value increases, although the minimum values remain comparable to those of 0.01 and 0.05. The superior performance with lower evaporation coefficients suggests that the ants require a certain degree of exploration before converging on the optimal solution. Figure 4.15b depicts the results for the ratio between the maximum and minimum pheromone bounds (τ_{ratio}). Larger values of τ_{ratio} extend the pheromone range. This favours the reinforcement of the

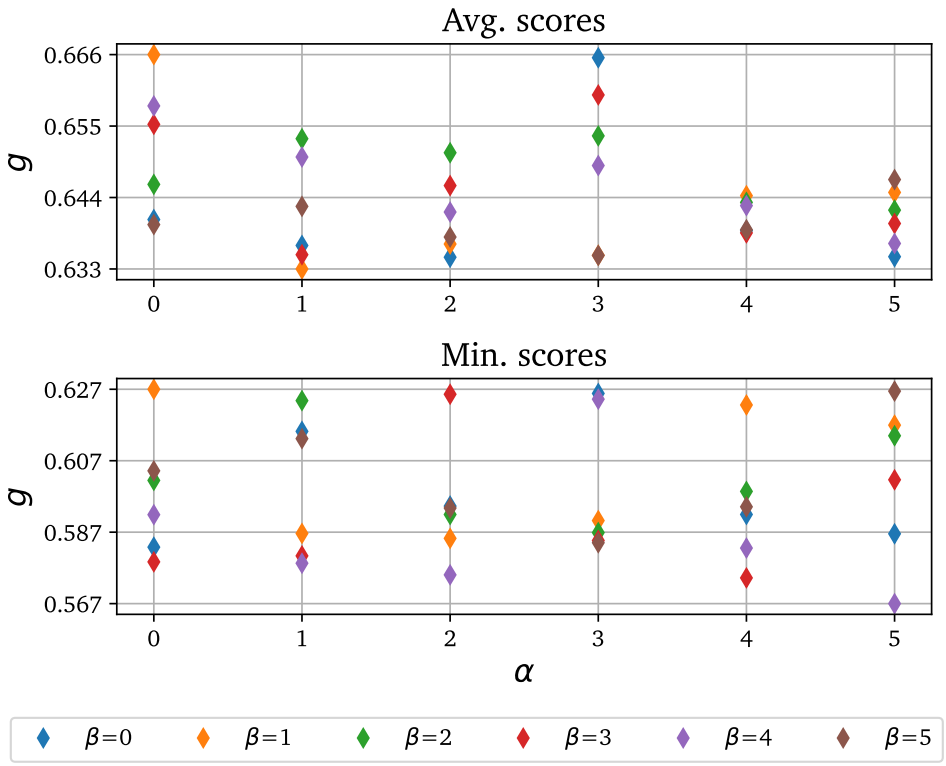


Figure 4.14: Results of the sensitivity on the (α, β) couple.

best solution and speeds up convergence but reduces exploration. Conversely, low values restrict the pheromone range, favouring exploration but slowing convergence. In this case, the average objective function value follows a parabolic trend with respect to τ_{ratio}, with the minimum occurring at $\tau_{\text{ratio}} = 20$. This value lies in the lower half of the range, confirming that the algorithm performs better when exploration is favoured over exploitation. Figure 4.15c shows the sensitivity to the number of iterations without improvement that triggers a restart (n_{imp}). The best value for both the average and the minimum is $n_{\text{imp}} = 30$. Similar to τ_{ratio}, the optimal value is found towards the lower end of the range where exploration is higher due to more frequent pheromone resets. Linked to n_{imp} is the memory factor (b_{res}), which determines how much of the old pheromone value is retained. When $b_{\text{res}} = 1$, memory is disabled and all pheromones are reset to the same value. Conversely, with $b_{\text{res}} = 0$, the restart does not affect the old pheromone levels. The memory factor yielding the best performance is $b_{\text{res}} = 0.9$, further demonstrating that the algorithm performs better when changes in pheromone levels are limited. However, results with $b_{\text{res}} = 1$ are inferior to those with lower memory factors such as 0.1 and 0.3. This indicates that retaining some memory of old pheromone values guides the ants to better solutions compared to a complete reset. Finally, the last parameter addressed is the switch frequency (n_s) between the global-best and iteration-best solutions used to update pheromones (Equation (4.17)). Recall that for the first 2000 iterations, only the iteration-best update is employed to avoid premature

convergence. The best values, $n_s = 30$ and $n_s = 50$, are roughly in the middle of the tested range, with $n_s = 30$ yielding slightly better results. In this context, predicting the effect of n_s is challenging. For instance, a high value of n_s promotes exploration during the iteration-best phase. However, once the switch to global-best occurs, the global solution is reinforced for n_s consecutive iterations, which increases exploitation.

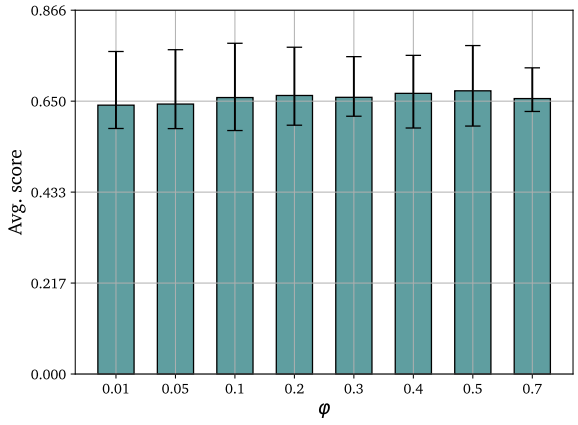
The sensitivity analysis demonstrates that the algorithm performs better when the ants are free to explore new solutions. This indicates that reinforcing the best solutions too heavily causes the algorithm to become trapped in local minima. To provide a quantitative assessment of parameter sensitivity, Table 4.5 lists the standard deviation of the average and minimum objective function values for each parameter variation. For the minimum values, the standard deviation lies between 1.2% and 1.7%. However, these minimum values represent single or rare occurrences over the 20 runs. The deviation of the average values offers a more consistent measure of parameter influence. The evaporation coefficient φ has the most significant impact. This parameter is used after every ant tour during the online local pheromone update and after each generation during the global pheromone update. The parameter with the least influence is the memory factor, as it is used only during restarts and its effect diminishes as iterations and pheromone updates continue. The standard deviations for the parameters are relatively low, demonstrating that the algorithm is quite robust. This robustness is partly due to the high number of iterations required for convergence, which gives the ants more opportunities to find better solutions and reduces their sensitivity to specific parameter settings. With a lower convergence criteria, the differences between parameter values would likely be more pronounced.

4.7. Evaluation of the start configuration

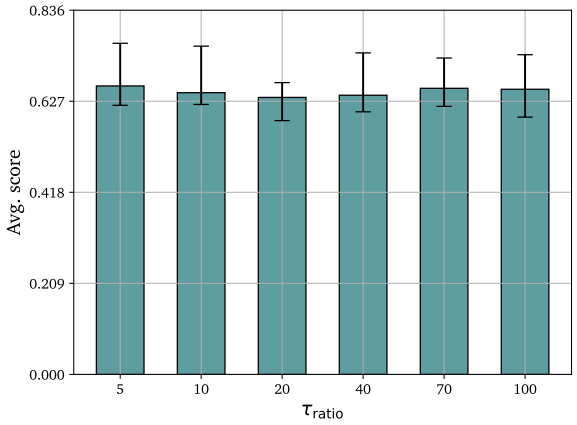
The initial configuration for the cooling channel is composed of two parallel standard serpentines (Figure 3.2a). To compare this initial configuration with the optimisation results, the former has been modeled using the same methodology described in Section

Table 4.5: Standard deviation between the average and minimum values obtained from the different parameter values tested.

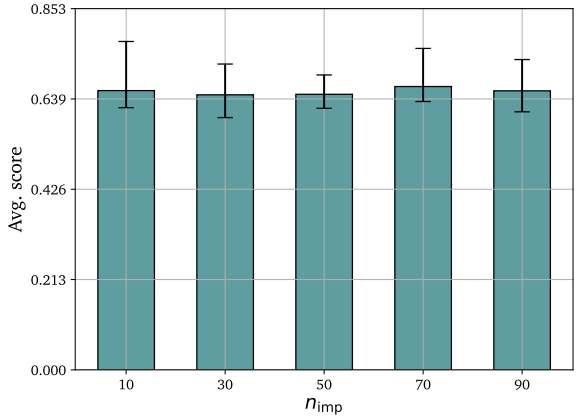
Parameter	$\sigma_{\text{avg vals}}$	$\sigma_{\text{min vals}}$
(α, β) couple	0.008	0.017
Evaporation coeff. φ	0.011	0.015
Max-min pheromones ratio τ_{ratio}	0.009	0.014
No improvement iter. n_{imp}	0.007	0.012
Restart memory factor b_{res}	0.0052	0.016
Switch frequency iter/global-best n_s	0.0075	0.014



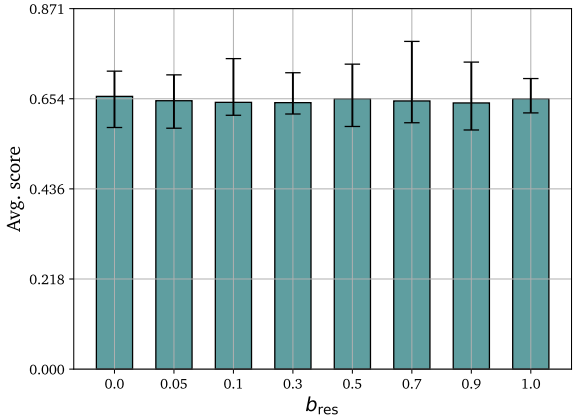
(a) Test on evaporation coefficient (φ).



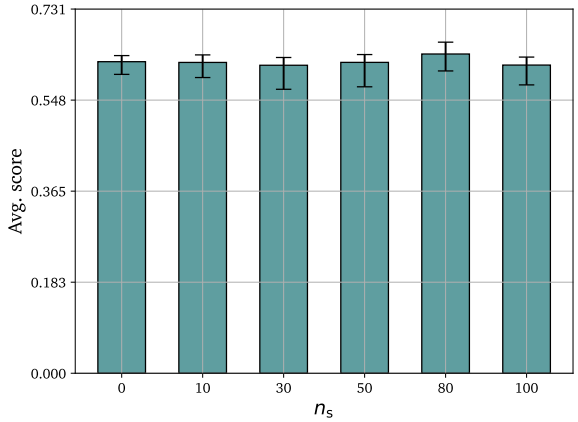
(b) Test on max-min pheromones ratio (τ_{ratio}).



(c) Test on number of iterations without improvement before restart (n_{imp}).



(d) Test on the memory factor used for restart (b_{res}).



(e) Test on number of iterations for the switch between the global-best and iter-best pheromone update (n_{s}).

Figure 4.15: Results of the sensitivity analysis. The histogram plots the average value between 20 runs, with also an indication on the maximum and minimum value.

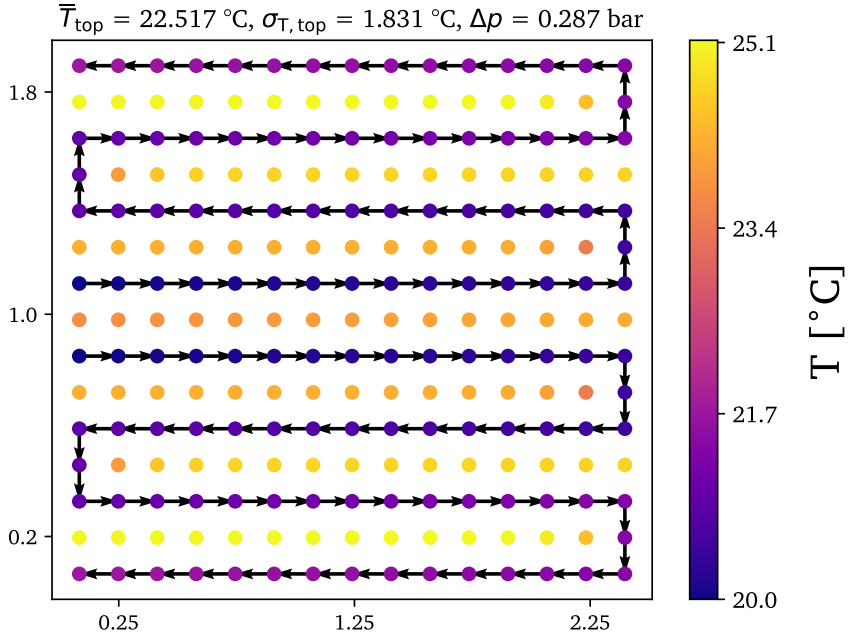


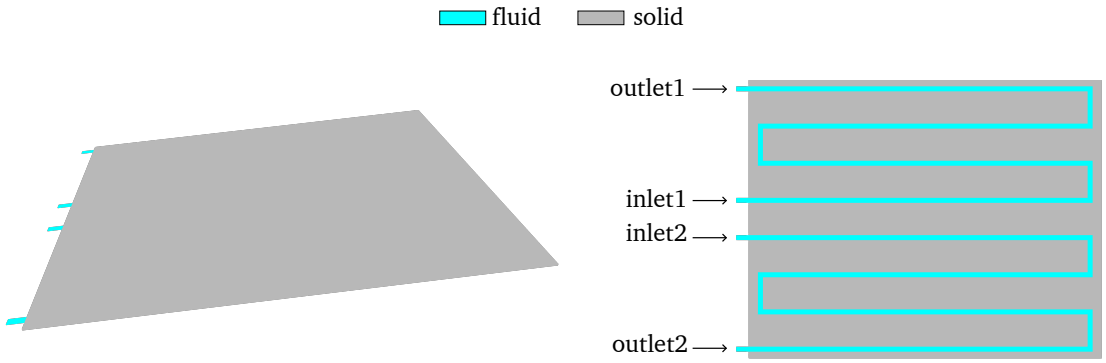
Figure 4.16: Equivalent thermal network of the initial configuration.

4.5, employing layers composed of a 15 × 15 grid. As a steady-state heat source, the mean heating load from the battery packs analysed in the previous chapter (Chapter 3) is imposed, corresponding to 2×10^3 W over the simulated time. The flow conditions and channel geometry remain identical to those used in the previous investigation (Table 3.2).

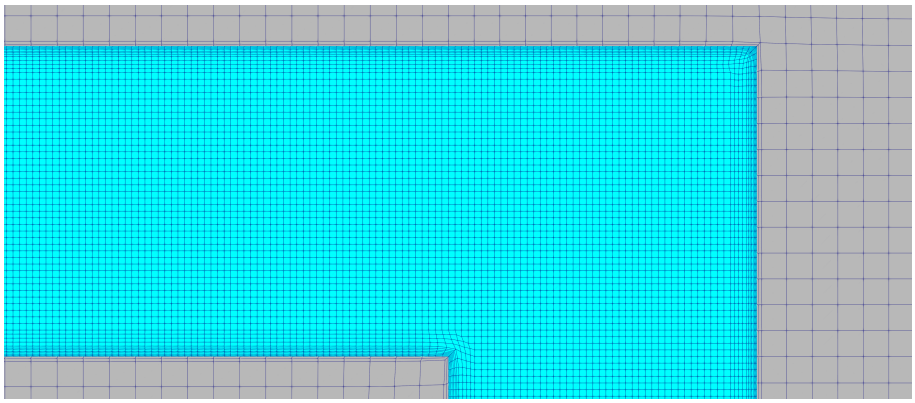
Figure 4.16 presents the results of the equivalent thermal network for the initial configuration. The plate features an average top surface temperature of 22.5 °C and a temperature standard deviation of 1.8 °C. The pressure drop between the global inlet and global outlet is $\Delta p = 0.29$ bar.

The initial configuration has also been simulated using a 3D CFD simulation to obtain high-fidelity results. This is particularly important for analysing temperature gradients, which may be altered in the LPTN approach due to the lower volume discretisation compared to CFD meshes. The CFD model is developed in OpenFOAM using the steady-state conjugate heat transfer solver `chtMultiRegionSimpleFoam`. The solved equations include the mass balance, momentum balance, and thermal energy balance. These equations, for the fluid domain, are shown in Equation (3.23), and heating due to viscous dissipation is neglected. In the solid domain is solved only the energy equation, which takes the form of the temperature Laplacian (Equation (3.24)).

To rapidly generate meshes for the additional configurations obtained by the optimisation procedure, an automatic meshing method has been developed, based on the node coordinates of the fluid channels and the OpenFOAM cartesian mesher `blockMesh`. Due to the difficulty of



(a) 3D view and slice at half height.



(b) Mesh at the interface between fluid and solid.

Figure 4.17: CFD geometry and solid-fluid interface of the initial configuration.

creating a proper boundary layer in the elbows (where the channel direction changes) using `blockMesh`, the boundary layer is added *a posteriori* using `SnappyHexMesh` by activating only the "add layers" step. Given the high number of cells required to mesh the channels and the entire solid region, the interface between the fluid and solid regions is non-conformal, as in [223]. This allows the solid region to be discretised with thicker cells compared to the fluid region, saving computational cost. It is important to note that only the Laplacian of the temperature is solved in the solid region; therefore, no convective terms are involved. Consequently, the mesh for the solid region does not require the strict constraints (such as orthogonality and low skewness) necessary for the fluid region.

Figure 4.17a illustrates the CFD domain of the initial configuration, including a slice at half-height to highlight the channel paths, which corresponds to the layout shown in Figure 4.16. The inlet and outlet sections are elongated to allow the flow to fully develop before entering or after exiting the cooling plate. Figure 4.17b depicts the non-conformal interface between the fluid and solid domains, showing the boundary layer in the fluid mesh at the elbow and the local refinement of the solid mesh at the interface. Due to the use of `snappyHexMesh` for generating the boundary layer, the cells at the corners exhibit a slight deviation from perfect orthogonality. However, this distortion does not affect the accuracy

of the simulation, as the mesh fully satisfies the quality criteria verified by the OpenFOAM `checkMesh` utility.

Table 4.6 lists the boundary conditions for the CFD model. The same velocity is imposed at both inlets, as the flow rate is perfectly halved due to the identical geometry of the channels. For other configurations where the channel geometry differs, the inlet velocity is extracted from the flow rate splitting algorithm applied in the LPTN model. The inlet temperature is set to 20 °C, and a zero-gradient condition is imposed on the pressure at the inlet. At the outlets, the pressure is set to 0.0, while a zero-gradient condition is employed for velocity and temperature. Since the flow is assumed to be incompressible, the absolute pressure value is not relevant; only the pressure difference between the inlet and outlet matters. Therefore, the value imposed at the outlet acts only as a reference value. At the fluid wall, a no-slip condition is imposed for velocity, and heat and temperature continuity are ensured between the fluid and solid. Finally, apart for the top solid wall where a uniform heat flux of 411 $[\mathrm{W\,m^{-2}}]$ is applied, all other walls are considered adiabatic.

As turbulence model has been employed the k–ω SST model, which, thanks to the blending between high-Re and low-Re approaches, is feasible also for transitory regimes. At the fluid walls, the turbulence is resolved with the low-Re approach. The pressure-velocity decoupling is managed by the SIMPLE algorithm [129], while the advective terms are resolved with the second-order upwind scheme. The total Reynolds number is $\mathrm{Re} = 6 \times 10^3$, thus in each channel the Reynolds is $\mathrm{Re} = 3 \times 10^3$, a value typical of transitional regime. This causes stability issues of the turbulence variables, for this reason only the turbulence variables advective terms are discretised with the first-order upwind. All the gradient are resolved with the second-order central difference scheme.

To assess the dependence of the results on the spatial discretisation, a mesh sensitivity analysis was performed on the geometry of the initial configuration. Three meshes were compared: the coarse mesh, with a mean cell size in the fluid region of 1.8 mm and a total of 1.8 million cells; the intermediate mesh, featuring a mean cell size in the fluid region of

Table 4.6: CFD simulation boundary conditions.

Boundary	Velocity	Pressure	Temperature
Inlet1/2	$0.921\,\mathrm{m\,s^{-1}}$	Zero-gradient	20.0 °C
Outlet1/2	Zero-gradient	0.0	Adiabatic
Fluid walls	No-slip	Zero-gradient	Heat and temperature continuity
Top solid wall	-	-	Imposed heat flux (q=411 $[\mathrm{W\,m^{-2}}]$)
Bottom solid wall	-	-	Adiabatic
Front solid wall	-	-	Adiabatic
Back solid wall	-	-	Adiabatic

1.4 mm and 3.3 million cells; and the fine mesh, with a mean cell size in the fluid region of 0.7 mm and 39 million cells. It should be noted that the refinement ratio of the mean cell size between the fine and intermediate meshes is 0.5, while between the intermediate and coarse grids the ratio is 0.78, as doubling the cell size of the intermediate mesh would result in insufficient resolution to discretise the solid layer between the channel top wall and the plate's top surface.

Table 4.7 summarises the results of the mesh sensitivity analysis. The table reports the cell count, the first cell height near the wall, the average y^+, the mean top wall temperature (including the deviation relative to the coarser grid), and the temperature standard deviation (also with the deviation relative to the coarser mesh). The parameters analysed show very small variations. This is expected for the comparison between the intermediate and coarse meshes, given the cell size ratio of 0.78. However, the results remain very close even when comparing the intermediate and fine meshes, despite the substantial difference in cell count. The largest variation is observed in the temperature standard deviation between the fine and intermediate meshes, which is −1.41%, whereas the mean temperature varies by only −0.06%.

It is worth noting that in all tested grids, the average dimensionless wall distance ($\bar{y}^+$) exceeds unity. Theoretically, to apply low-Re wall treatment, the condition $y^+ \leq 1$ should be satisfied. Reducing $\bar{y}^+$ would require decreasing the height of the first cell adjacent to the wall (Δ_{wall}), but reducing this size would lead to an exponential increase in the total cell count, which is already computationally demanding across all the three grids. Nevertheless, for many standard applications, a value of $y^+ < 5$ is considered acceptable and this threshold is fully respected even by the coarse mesh.

Based on the grid sensitivity analysis, it is concluded that the intermediate mesh is sufficient. It yields results very close to the fine mesh for the parameters of interest while significantly reducing the computational burden.

Finally, a comparison between the results of the LPTN model and the CFD simulation with the fine mesh is presented in Table 4.8. The table lists the temperature difference between the mean temperature of the top wall and the coolant inlet temperature, the standard deviation of the top wall temperature, and the pressure drop. The relative error for the LPTN model with respect to the CFD model is calculated for each metric.

Table 4.7: Results of the mesh sensitivity analysis on the initial configuration.

Grid	n_{cells}	Δ_{wall} [m]	$\bar{y}^+$	$\overline{T}_{\text{top}}$ [°C]	$\overline{T}_{\text{top}}$ diff.	$\sigma_{\text{T,top}}$ [°C]	$\sigma_{\text{T,top}}$ diff.
Coarse	1.8×10^6	2.8×10^{-4}	3.8	22.30	-	0.69	-
Intermediate	3.3×10^6	2.1×10^{-4}	2.76	22.41	−0.04%	0.70	−1.21%
Fine	39×10^6	1.2×10^{-4}	1.60	22.60	−0.06%	0.71	−1.41%

The LPTN model predicts the mean temperature of the top wall with an accuracy of 3.16% relative to the CFD simulation. However, it fails to accurately capture the standard deviation of the temperature, overestimating it by 157% compared to the CFD results. This discrepancy is primarily due to the fundamental differences in the modeling approaches. In the LPTN model, the conductive resistances connecting the solid nodes are calculated based on the distance between nodes. However, since each node lumps a significant volume, this simplified resistance network assumes heat travels the full distance between centers, which can artificially increase thermal resistance and isolate nodes. The Finite Volume Method employed in CFD simulations also assumes constant temperature within a cell volume, but the spatial discretisation in the CFD model is orders of magnitude finer, effectively almost canceling this discretisation error and capturing continuous thermal gradients.

Additionally, in the LPTN model, solid nodes in the top layer exchange heat laterally only with their four immediate neighbours, and heat transfer with the channel layer below is modelled by a single vertical resistance. This coarse connectivity limits the simulation of heat spreading in diagonal direction within the solid. Furthermore, the convective heat transfer in the LPTN is calculated using correlations that assume fully developed flow, whereas the CFD simulation resolves complex flow features such as recirculation and secondary flows that enhance local mixing. These factors make the LPTN model feasible for calculating mean temperatures but less accurate for predicting local temperature gradients. However, since the objective of the present optimisation is to improve the temperature uniformity, this error is accepted, provided that a relative improvement in the LPTN model corresponds to a qualitative improvement in the high-fidelity CFD model.

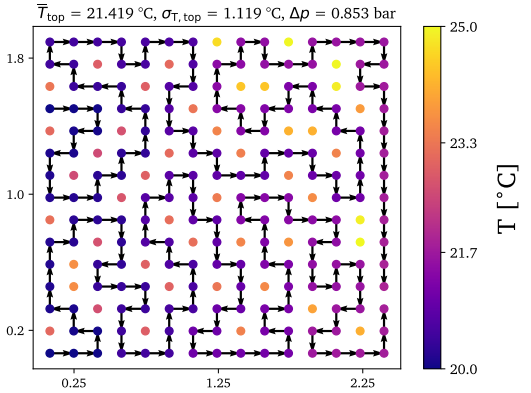
Finally, the comparison of pressure losses shows an overestimation by the LPTN model of 1.3%. This results confirms the effectiveness of the flow rate splitting algorithm, which is based on the calculation of head losses.

4.8. Optimisation result

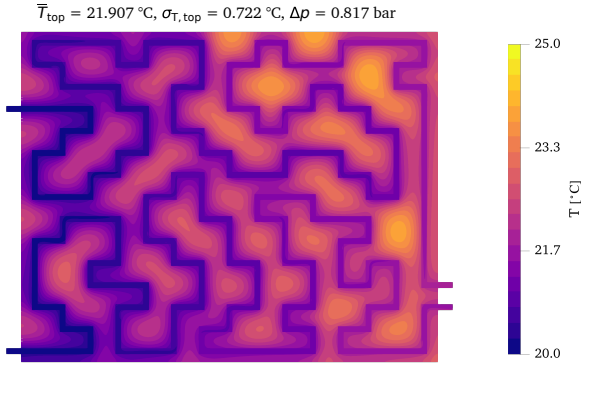
Based on the sensitivity analysis of the parameters, the optimal choice for the grid and channel configuration is a 15×15 grid featuring two channels. The parameters used

Table 4.8: Comparison between the CFD model with fine mesh and LPTN model of the initial configuration.

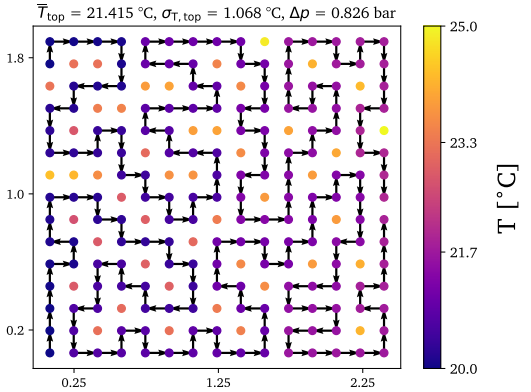
Parameter	CFD	LPTN	LPTN err.
$\overline{T}_{\text{top}} - T_{\text{in}}\,[^\circ\text{C}]$	2.60	2.52	3.16%
$\sigma_{\text{T,top}}\,[^\circ\text{C}]$	0.71	1.83	157.60%
$\Delta p\,[\text{Pa}]$	0.287×10^5	0.29×10^5	1.3%



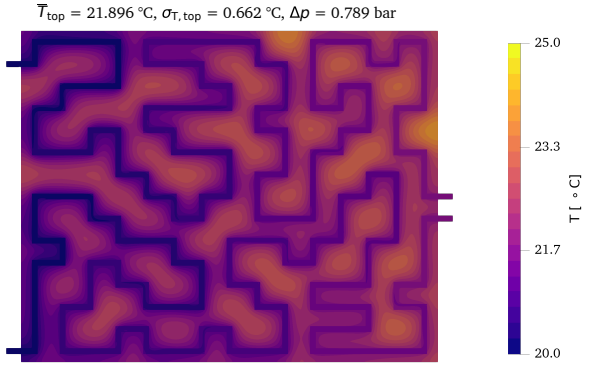
(a) Best solution with ($\alpha = 1$, $\beta = 1$).



(b) CFD temperature map of the best solution with ($\alpha = 1$, $\beta = 1$).



(c) Best solution with ($\alpha = 5$, $\beta = 4$).



(d) CFD temperature map of the best solution with ($\alpha = 5$, $\beta = 4$).

Figure 4.18: Best solutions obtained from the optimisation.

correspond to those listed in Table 4.2, with the exception of the number of iterations between switching from the reinforcement of the iteration-best solution to the reinforcement of the global-best solution. The value used in the sensitivity test was $n_s = 50$; however, the analysis in Figure 4.15e (completed after the grid size sensitivity analysis) indicated that $n_s = 30$ yields slightly better performance. Regarding the exponents for the pheromones and the heuristic function (α, β), two different parameter pairs were chosen: the pair yielding the minimum average score ($\alpha = 1$, $\beta = 1$) and the pair yielding the absolute minimum score ($\alpha = 5$, $\beta = 4$). For each set of parameters, 20 independent runs were performed, resulting in a total of 40 runs.

Table 4.9 compares the results obtained using the different exponent pairs. The differences in the minimum, average, and standard deviation of the scores are small, with the pair ($\alpha = 5$, $\beta = 4$) performing slightly better across all metrics. It is worth noting that in the analysis presented in Figure 4.14, the minimum average (0.633) was obtained with ($\alpha = 1$, $\beta = 1$),

Table 4.9: Comparison of the results of the algorithm with ($\alpha = 1$, $\beta = 1$) and ($\alpha = 5$, $\beta = 4$).

(α, β)	g_{min}	g_{avg}	g_{dev}
(1,1)	0.518	0.574	0.0143
(5,4)	0.511	0.568	0.0133

whereas the pair ($\alpha = 5$, $\beta = 4$) was very close (0.636). Regarding the absolute minimum, the results are consistent with the sensitivity analysis, where the pair ($\alpha = 5$, $\beta = 4$) performed better. The difference between the minimums obtained with the two pairs is more pronounced than the difference between their average values. In this context, it appears that while the pheromones and the heuristic information have similar importance, the pheromone trail should carry slightly more weight than the heuristic. Additionally, higher values for α and β will be tested, as results could potentially improve further.

In both cases, all obtained solutions respect the constraints; thus, the penalisation factor was never applied.

Figures 4.18a and 4.18c display the two best results obtained with ($\alpha = 1$, $\beta = 1$) and with ($\alpha = 5$, $\beta = 4$), respectively

As previously discussed regarding the node count sensitivity analysis, the optimal solutions are those where the distribution of fluid nodes covers nearly all regions of the plate, incorporating more turns to increase the channel length and, consequently, the convective heat transfer area. Both these aspects are present in the solutions shown in Figure 4.18; additionally, the number of fluid nodes (and solid nodes) is identical between the two solutions. The results are very similar, but the configuration in Figure 4.18c exhibits both a lower mean temperature and a lower standard deviation for the top layer. Furthermore, the pressure losses are lower as the latter presents fewer changes of direction (122 in Figure 4.18a vs 116 in Figure 4.18c).

To investigate the differences between the two solutions quantitatively, three different metrics, considering the interactions between the solid and fluid nodes in the channels layer, were formulated:

i. The first metric is the convective resistance density, calculated as the ratio between the number of resistances linking a fluid and a solid node ($N_{R_{\text{f-s}}}$) and the total number of resistances in the channels layer ($N_{R_{tot}}$):

$$R_{\text{conv,ratio}} = \frac{N_{R_{\text{f-s}}}}{N_{R_{tot}}} \,. \tag{4.57}$$

ii. The second metric attempts to quantify the proportion of potential convective heat

transfer resistances that are utilised. For each solid node, this is calculated as the ratio between the number of fluid neighbours (N_{fn}) and the number of potential neighbours ($N_{\mathrm{n}} = 4$ at the centre, $N_{\mathrm{n}} = 3$ at the boundaries, and $N_{\mathrm{n}} = 2$ at the corners). The mean over all solid nodes is then computed:

$$\overline{\Psi} = \frac{\sum_{i}^{N_{\mathrm{solid}}} \frac{N_{\mathrm{fn,i}}}{N_{\mathrm{n,i}}}}{N_{\mathrm{solid}}} \,. \tag{4.58}$$

iii. The third metric assesses the average effectiveness of the solid-fluid interactions by calculating the average fluid temperature with which a generic solid node interacts. This is achieved by iterating over all solid nodes in the channels layer; for each fluid neighbour of a solid node, the temperature difference with respect to the inlet temperature is calculated ($\Delta T_{\mathrm{fluid}} = T_{\mathrm{fluid}} - T_{\mathrm{in}}$). These differences are summed and then averaged over the total number of fluid-solid connections:

$$\overline{\gamma} = \frac{\sum_{i}^{N_{\mathrm{solid}}} \left[\sum_{j}^{N_{\mathrm{fn,i}}} (T_{\mathrm{j}} - T_{\mathrm{in}}) \right]}{N_{R_{\mathrm{f-s}}}} \,. \tag{4.59}$$

Table 4.10 summarises the defined metrics for the best solution with ($\alpha = 1$, $\beta = 1$) (Figure 4.18a), the best solution with ($\alpha = 5$, $\beta = 4$) (Figure 4.18c) and the initial configuration (Figure 4.16). Comparing the two optimised solutions, both $R_{\mathrm{conv,ratio}}$ and $\overline{\Psi}$ are slightly higher in the first case, indicating that the first solution features a higher number of interactions between fluid and solid nodes. However, $\overline{\gamma}$ is lower (better) in the second case, indicating that, on average, the solid nodes interact with fluid neighbours at lower temperatures compared to the first case. This can be confirmed qualitatively by examining Figures 4.18a and 4.18c. It can be observed that in the second case, the clusters of solid nodes are located closer to the inlets where the fluid is at a lower temperature. The mean x coordinate (indicating the column position) of the solid nodes in the first case is $\overline{x}_{\mathrm{solid}}$=1.15 m, whereas in the second case it is $\overline{x}_{\mathrm{solid}}$=1.08 m. This demonstrates that in the second case, the center of mass of the solid regions is shifted towards the inlets.

To summarise, the solution with ($\alpha = 1$, $\beta = 1$) features slightly more interactions between fluid and solid nodes, but in the solution with ($\alpha = 5$, $\beta = 4$), these interactions occur at a lower fluid temperature, resulting in a lower mean temperature and better temperature uniformity. However, this observation requires verification with CFD simulations to confirm that the second solution is indeed superior, particularly given the small magnitude of the difference.

Regarding the initial configuration, the higher number of solid nodes and shorter fluid paths lead to a higher density of links between solid and fluid nodes, while the number of

fluid branches is lower. Looking at $\overline{\Psi}$, it becomes evident that the distribution of fluid and solid nodes in the initial configuration is far from optimal, since on average only half of the possible connections with solid nodes are utilized as convective heat transfer resistances. In fact, the initial configuration features large continuous solid regions connected by conductive resistances, which also leads to a higher mean fluid temperature with which a generic solid node interacts.

The 3D CFD models for the two solutions have been developed following the methodology used for the initial configuration, maintaining the intermediate mesh size and the same simulation settings. Figures 4.18b and 4.18d show the mid-plane temperature maps resulting from the CFD simulations of both solutions. It can be noted that the hotspots identified in the CFD simulations are correctly represented in the equivalent thermal network solutions. Furthermore, as hypothesised from the reduced-order models, the more continuous solid mass near the outlet section in the configuration with ($\alpha = 1$, $\beta = 1$) leads to a slightly higher temperature compared to the case with ($\alpha = 5$, $\beta = 4$).

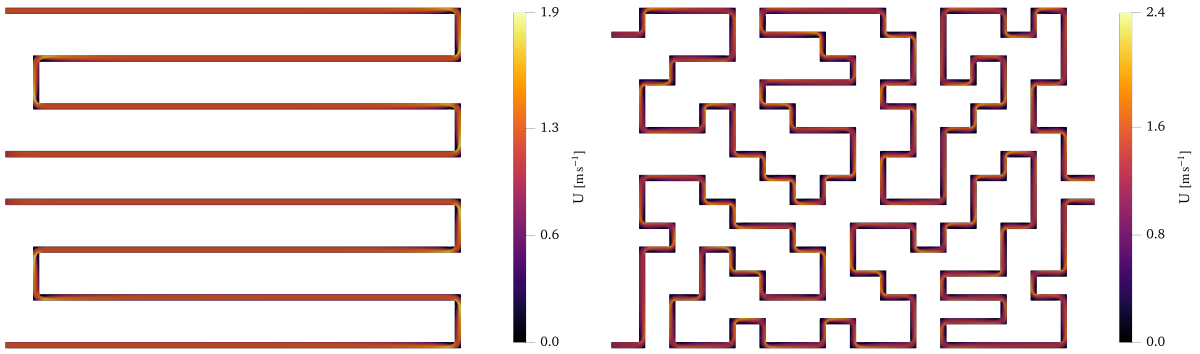
Table 4.11 compares the solutions obtained with the LPTN and CFD models for the best configurations obtained with ($\alpha = 1$, $\beta = 1$) and ($\alpha = 5$, $\beta = 4$). The CFD simulations confirm that the best solution, obtained with ($\alpha = 5$, $\beta = 4$), leads to a lower mean temperature of the plate's top wall, a lower temperature standard deviation, and lower pressure losses. The trend predicted by the LPTN model is therefore in accordance with the CFD results. However, when compared to the analysis of the initial configuration (Table 4.8), the LPTN model exhibits a larger error in the mean temperature difference (approximately 25% vs. 3.16%). This discrepancy can be attributed to the more complex flow conditions present in the optimised solutions, as the LPTN model employs a correlation that assumes fully

Table 4.10: Comparison of the defined metrics between the best solutions obtained with ($\alpha = 1$, $\beta = 1$), ($\alpha = 5$, $\beta = 4$) and the initial configuration.

Solution	$R_{\text{conv,ratio}}$	$\overline{\Psi}$	$\overline{\gamma}$
Figure 4.18a	0.41	0.95	0.82
Figure 4.18c	0.40	0.92	0.79
Figure 4.16	0.49	0.53	0.84

Table 4.11: Comparison between the CFD and LPTN model of the best solutions.

	($\alpha = 1$, $\beta = 1$)			($\alpha = 5$, $\beta = 4$)		
Parameter	CFD	LPTN	LPTN err.	CFD	LPTN	LPTN err.
$\overline{T}_{\text{top}} - T_{\text{in}}$ [°C]	1.907	1.419	26%	1.896	1.415	25%
$\sigma_{\text{T,top}}$ [°C]	0.722	1.119	55%	0.662	1.068	61%
Δp [Pa]	0.817×10^5	0.853×10^5	4%	0.789×10^5	0.826×10^5	5%



(a) Velocity magnitude of the initial configuration. **(b)** Velocity magnitude of the solution with ($\alpha = 5$, $\beta = 4$).

Figure 4.19: Velocity magnitude from the CFD simulations.

developed flow conditions for the convective heat transfer coefficient. While this assumption is generally valid for the initial configuration, except near the corners, it is rarely applicable to the optimised layouts, where frequent changes of direction prevent the flow from fully developing. The difference in flow conditions can be visualised in Figure 4.19. Figure 4.19a shows the velocity magnitude map of the initial configuration, while Figure 4.19b depicts the velocity magnitude of the best solution obtained from the optimisation with ($\alpha = 5$, $\beta = 4$). Comparing the two colour maps it can be noted that in the initial configuration, the flow is fully developed in the long horizontal branches, which represent the largest portion of the channels. Conversely, the optimised configuration features a high number of curves that interrupt the flow development in the straight segments. This difference is also reflected in the head losses, which increase by a factor of 2.72 in the optimised case.

The error in the temperature standard deviation predicted by the LPTN model decreases for the optimised configurations. This reduction can be attributed to the reduction of isolated solid nodes in the optimised paths compared to the initial geometry, which mitigates the error in computing temperature gradients inherent to the LPTN approach. The error in the pressure drop is larger for the new configurations, although it remains low in absolute value ($\leq 5\%$). This increase is due to the higher number of direction changes in the optimised layouts, which increases the relative weight of the concentrated pressure losses compared to the initial configuration. It is recalled that in the LPTN model, the pressure drops at the corners are modeled as concentrated losses calculated using a constant coefficient of $K = 2.0$, as indicated in [171].

Table 4.12 compares the plate's top surface mean temperature, the temperature standard deviation and the pressure drop for the initial configuration and the best solution obtained from the optimisation, which is the one with ($\alpha = 5$, $\beta = 4$). The columns labeled with ε indicate the ratio between the optimised and initial values. Taking the results from the CFD approach as a reference, the improvement in the plate's top surface mean temperature is

27%, corresponding to a ratio of ε_{CFD}=0.73. The improvement in the temperature standard deviation is 7%. The improvement in the thermal properties is mostly due to the increased convective heat transfer area and the better distribution between the fluid and solid domains. Another factor which contributes to the improvement is the higher number of flow obstacles in the optimised solution, which accelerates the fluid enhancing the local heat transfer coefficient, although this effect is not accounted for in the LPTN model used in the ACO evaluation procedure. As already stated, the head losses increase by a factor of 2.72, remaining under the constraint of 1×10^5 Pa. The LPTN model correctly predicts the trend of the ratios. However, for the reasons previously mentioned, the specific values of the ratios for the mean temperature and standard deviation present some differences compared to the CFD reference, while the ratio of the pressure drop is very similar to the one predicted by the CFD simulations.

4.9. Concluding remarks

This chapter proposes a novel meta-heuristic methodology for the reduced-order optimisation of cooling plate channels. The addressed optimisation is based on the ACO algorithm, a meta-heuristic approach inspired by the foraging behaviour of ant colonies. In nature, ants explore the surroundings of their nest to locate food sources. Upon finding food, they return to the nest, depositing a pheromone trail, *i.e.* a chemical substance used to communicate the route to the food source to the rest of the colony. Shorter paths allow ants to complete more trips in a given time, leading to an accumulation of pheromones. This higher pheromone concentration attracts more ants, which further contribute to the reinforce the trail.

The ACO algorithm was introduced in 1992, and its first application was the Traveling Salesman Problem (TSP), given the analogy with ants searching for the shortest path to a food source. In the TSP, the objective is to visit a given set of cities while minimising the total travelled distance. This direct correspondence made the initial adaptation of the ACO algorithm straightforward. However, when applied to large sets of cities or more complex problems, the limitations of the ACO are more apparent. Because of the strong positive feedback mechanism, the algorithm can easily converge to local minima. This behaviour prompted the development of various ACO variants, aimed at improving the balance between

Table 4.12: Comparison of the initial configuration and the best optimised solution using the CFD and LPTN models.

	Initial		Optimised			
Parameter	CFD	LPTN	CFD	ε_{CFD}	LPTN	$\varepsilon_{\text{LPTN}}$
$\overline{T}_{\text{top}} - T_{\text{in}}$ [°C]	2.60	2.52	1.896	0.73	1.415	0.56
$\sigma_{\text{T, top}}$ [°C]	0.71	1.83	0.662	0.93	1.068	0.58
Δp [Pa]	0.287×10^5	0.290×10^5	0.789×10^5	2.72	0.826×10^5	2.88

the exploitation of existing knowledge and the exploration of new solutions.

Due to the differences between the TSP and the optimisation of cooling channel paths, the proposed algorithm incorporates several modifications to the classical ACO framework. These changes are necessary to enhance exploration and respect physical constraints. The solid and fluid domains of the cooling plate are discretised using a grid of nodes, where each node represents a specific discrete volume of the plate. Consequently, unlike the TSP, the fluid paths must be continuous, must navigate around solid obstacles, and must not intersect when multiple parallel channels are generated. Key features added to the standard ACO include the re-initialisation of pheromone levels, the mutation of the current best solution, and the capability for a single ant to construct multiple parallel paths.

The solutions generated by the ants are evaluated using an equivalent LPTN, which is resolved by the solver detailed in Section 2.7. The convective heat transfer coefficient required for the thermal network is calculated using an empirical correlation. Additionally, the solver integrates a procedure to calculate the flow rate distribution when multiple parallel channels are present.

The proposed algorithm was applied to optimise the cooling plate described in Chapter 3, which is designed to cool two battery packs and the power converter of a BEV. Given the large number of parameters involved in ACO-based algorithms, a sensitivity analysis of the main parameters was performed prior to the final optimisation. For each parameter set, 20 independent runs are executed to collect statistical data. This statistical approach is necessary because the heuristic nature of the ACO can lead to converge to local minima rather than the global optimum. Nevertheless, due to the low computational cost and the high potential for parallelisation, executing multiple optimisation runs does not present a big issue.

Having identified the optimal set of parameters, the ACO algorithm was employed to find the best configuration for the specific cooling plate. The objective was to minimise the temperature of the plate's top surface, which is in direct contact with the batteries, and to improve the overall temperature uniformity. These improvements are intended to ensure appropriate operating conditions for the battery cells and mitigate degradation. Within the optimisation algorithm, the constraints regarding the maximum temperature standard deviation and the maximum pressure losses are implemented as penalty factors. This prevents discarding solutions that present minor constraint violations. A slight violation of a constraint within the LPTN model might not correspond to a violation in the physical system. This is particularly true for the temperature standard deviation, given the inherent limitations of the lumped parameter approach in resolving local thermal gradients.

Compared to the initial layout, the optimised solution features longer channels, resulting in a larger convective heat transfer area and a more effective spatial distribution of the solid and fluid nodes. To achieve this configuration, the fluid paths undergo more direction

changes rather than proceeding directly towards the outlets, which consequently increases the pressure losses. Furthermore, the frequent direction changes cause local fluid accelerations, which locally enhance the convective heat transfer coefficient. The initial and optimised solutions were also compared using 3D CFD simulations. This method provides more accurate results compared to the LPTN model, particularly concerning the temperature gradients. According to the CFD results, the optimised configuration reduces the difference between the top surface mean temperature and the coolant inlet temperature by 27%, while the temperature standard deviation is reduced by 7%. Finally, the pressure drop increases by a factor of 2.72, while remaining within the established constraint.

The proposed algorithm offers a computationally inexpensive and numerically robust alternative to higher-order topology optimisation methods based on CFD or FEM frameworks, which typically entail high computational costs and can suffer from numerical instabilities. Nonetheless, higher-order methods can achieve superior levels of optimisation due to their detailed domain representation and the application of gradient-based optimisation procedures. However, the solutions generated by higher-order methods are often difficult to manufacture using standard tools, frequently requiring the use of additive manufacturing technologies. Conversely, the proposed algorithm relies on a simpler domain representation and always returns geometries that can be easily manufactured. These results can also serve as an optimised initial configuration for subsequent higher-order optimisations.

This algorithm represents a novel approach, thus further detailed investigations are required to improve the performance. Future work will extend the sensitivity analysis of the main parameters and test new heuristic functions. The geometric optimisation of the channel aspect ratio will also be included, alongside the optimisation of the number of parallel channels, which is a fixed parameter in this initial version.

CHAPTER 5

Multiscale methodology for statistical analysis of infection risk from respiratory droplets

5.1. Introduction

The transition to electric mobility places a strong emphasis on energy efficiency to maximise vehicle driving range. In this context, the most significant auxiliary consumer is the Heating Ventilation and Air Conditioning (HVAC) system [224], which can reduce electric vehicle range by up to 45% in cold weather conditions [225]. Consequently, modern control logic often aims to reduce the thermal load from the outside air, leading to reduced fresh air intake rates or optimised recirculation strategies to maintain cabin comfort with lower energy consumption [226]. However, while these measures are effective for energy conservation, they may alter the cabin's air quality and safety. Muratori *et al.* [227] proposed a recirculation strategy based on the CO_2 level inside the cabin. This new strategy allowed for energy savings of 72.6% under heating conditions and 9.0% under cooling conditions, while maintaining low CO_2 levels. To identify the best cabin ventilation strategy, it is essential to understand how these environments affect the dispersion of airborne particles, such as exhaled respiratory droplets. As demonstrated by Arpino *et al.* [228], the dispersion of droplets in a confined car cabin is a non-isothermal phenomenon heavily influenced by the HVAC system's operation. Their work highlights that neglecting thermal effects, such as the buoyancy of warm exhaled air and the thermal plumes generated by passengers, leads to inaccurate predictions of droplet trajectories. Consequently, the assessment of infection risk cannot be decoupled from the detailed analysis of the cabin's thermal field, which is regulated by the same system responsible for battery thermal management and cabin climatisation. In

NOMENCLATURE

Latin letters

$\mathscr{B}$	Droplet size PDF [m^{-1}]
C_d	Drag coefficient
c	Concentration [m^{-3}]
D	Diameter [m]
$\mathscr{D}$	Drag force [N]
e	Viral exposure [$s\,m^{-3}$]
g	Gravitational acceleration [$m\,s^{-2}$]
h	Specific enthalpy [$J\,kg^{-1}$]
i	Viral inhalation
$\mathscr{I}$	Infectious dose PDF
J	Insulation coefficient [$m^2\,K\,W^{-1}$]
K	Evaporation rate [$m^2\,s^{-1}$]
k	Turbulent kinetic energy [$m^2\,s^{-2}$]
k'	Kinetic energy per unit mass [$m^2\,s^{-2}\,kg^{-1}$]
m	Generic parameter
N	Number
p	Pressure [Pa]
Q	Quanta exhaled
q''	Heat flux [$W\,m^{-2}$]
r	Risk of infection
r'	Radius, radial distance [m]
S	Surface area [m^2]
$\boldsymbol{s}$	Position [m]
Sc	Schmidt number
T	Temperature [K]
t	Time [s]
$\boldsymbol{u}$	Velocity [$m\,s^{-1}$]
V	Volume [m^3]
$\mathscr{W}$	Weight force [N]
x	Horizontal direction [m]
Y	Water mass fraction

Greek letters

α	Thermal diffusivity [$m^2\,s^{-1}$]
β	Entrainment coefficient
ε	Turbulent dissipation rate [$m^2\,s^{-3}$]
ζ	Non-dimensional radial distance
$\boldsymbol{\eta}$	Random vector
θ	Exhaled flow direction [rad]
μ	Dynamic viscosity [$kg\,m^{-1}\,s^{-1}$]
ρ	Density [$kg\,m^{-3}$]
τ	Characteristic time [s]
ω	Turbulent specific dissipation rate [$1\,s^{-1}$]

Subscripts

ab	Airborne
avg	Average
b	Reference scenario
c	Cell
cl	Clothes
cr	Critical
D	Exhaled droplets
d	Drag
e	Evaporation
eff	Effective
ex	Exhalation
gr	Ground
in	Inhalation
m	Humid air mixture
max	Maximum
min	Minimum
mt	Mouth or nose orifice
p	Droplet or particle
pk	Peak
pr	Period
r	Relative
rm	Room
sk	Skin
t	Terminal condition
V	Exhaled viruses
0	Initial condition

this context, numerical approaches are often preferred over purely experimental methods for detailed risk assessment. While experimental setups are essential for validation, they face inherent difficulties in capturing the full volumetric evolution of the exhaled breath cloud or distinguishing between varying particle sizes in real-time. Computational simulations,

conversely, allow for the precise tracking of individual droplets and their size distributions. Furthermore, numerical frameworks offer significant flexibility, enabling researchers to easily switch the analysis between different environmental conditions and ventilation strategies without the constraints of physical setups. However, the implementation of such numerical models requires a rigorous characterisation of the respiratory events. This involves defining the breathing activity from a fluid-dynamic perspective to capture the jet properties, as well as accurately specifying the release mechanisms and size distributions of the generated droplets.

5.2. Literature survey

The passenger car cabin is characterised by a small volume, high occupant density, and complex flow topology driven by dashboard vents and extraction outlets. Unlike larger indoor spaces, the proximity of occupants in a car precludes effective social distancing, making ventilation efficiency the dominant factor in reducing viral exposure.

Recent computational studies have addressed the complexity of airflow patterns within vehicle cabins to evaluate passenger safety. A significant contribution to this field is the work by Arpino *et al.* [228], who developed a transient non-isothermal 3D numerical model to analyse aerosol dispersion in a passenger car cabin. Their approach is based on a non isothermal Eulerian-Lagrangian framework implemented in the open-source software OpenFOAM. In this model, the airflow is treated as a continuous phase solving the Navier-Stokes and energy equations, while droplets are tracked as a discrete phase using Newton's equations of motion. The authors compared their simulations with Particle Image Velocimetry (PIV) measurements obtained from a scale model of a car cabin. Through this comparison, they evaluated various turbulence models, ultimately determining that the Shear Stress Transport (SST) k-ω model provided the most accurate prediction of the airflow patterns. This validation phase demonstrated the necessity of using refined computational grids to correctly capture the velocity gradients in the jet regions of the air vents. Building upon this validated numerical framework, the authors subsequently applied the model to a specific risk assessment of SARS-CoV-2 transmission [229]. In this second study, the CFD model was integrated with a predictive emission-to-risk model that accounts for viral load and dose-response dynamics. The investigation covered various scenarios, analysing the influence of the HVAC airflow rate, ventilation mode, passenger position, and the type of expiratory activity (breathing vs. speaking). The findings revealed that the infection risk is highly sensitive to the specific airflow patterns created by the HVAC system rather than just the total air exchange rate. For instance, the authors observed that while higher flow rates generally reduce risk, specific configurations like the windshield defrosting mode can drive droplets from the driver directly into the rear compartment, significantly increasing the exposure of rear seat passengers. Conversely, a front ventilation mode was found to create an aerodynamic barrier that effectively

isolates the rear passengers from the driver. Furthermore, the study compared these detailed 3D results with simplified zero-dimensional (well-mixed) models. The comparison showed that simplified models often fail to capture local high-risk zones and are only reliable under conditions of very low ventilation where the air is stagnant. Mathai *et al.* [230] conducted steady-state isothermal simulations to investigate the influence of car windows opening on the droplets dispersion. They investigate only the spread of airborne particles, *i.e.* dried particles transported by the local air-flow. The particles are treated like a continuous passive scalar, to mimic the spread of a tracer. Their simulations revealed that opening windows creates cross-flows that isolate the driver from the passenger more effectively than mechanical ventilation alone. However, they also noted that specific open window combinations can inadvertently draw contaminants from the passenger zone into the driver's breathing zone due to recirculation eddies. Sarhan *et al.* [231] used an Eulerian-Eulerian CFD model to investigate the dynamic of the infection in a car with 4 passengers due to breathing and speaking activities. Their results indicated that in a poorly ventilated cabin, the threshold for infection could be reached in approximately 6 minutes. Furthermore, they observed that increasing the HVAC air velocity effectively dilutes the droplet concentration, significantly delaying infection time. Regarding seating arrangements, for the specific position of the vents, the study highlighted that the passenger sitting directly behind the infected individual typically faces the highest exposure due to airflow patterns, while the front passenger seat was identified as the safest location when the driver is the potential source of infection. As for the car cabin, these approaches are used for the risk analysis of other indoor environments. Ramajo and Corzo [232] applied a non-isothermal Eulerian-Lagrangian CFD framework to investigate airborne transmission in urban buses, explicitly accounting for occupant thermal plumes. Their analysis distinguished between large ballistic droplets (>200 μm), which travel independently of the airflow, and smaller aerosols (<5 μm), which are heavily dispersed by the HVAC system. They found that while increasing air renewal rates is effective in reducing the overall infection risk, the airflow patterns generated by the HVAC system can lead to non-uniform droplet dispersion, potentially creating local zones of higher exposure depending on occupant seating and vent configuration. Yang *et al.* [233] also employed the same approach for the risk analysis in urban buses, focusing also on the role of relative humidity (RH). The study revealed that low RH conditions accelerate droplet evaporation into suspended nuclei, thereby facilitating long-range airborne transmission; conversely, high RH (95%) suppresses evaporation, causing droplets to retain their mass and settle rapidly due to gravity, which limits airborne spread but increases surface deposition. All these studies relied on CFD simulations due to their capability to model the complex airflow of breathing events and determine the spatial spread of the virus. However, a limitation is that CFD simulations are case-specific and computationally expensive. Conversely, other works in the literature propose analytical models to offer a more generalised approach. These

models solve the governing equations of droplet transport, energy balance, evaporation, and chemical composition, often expanding the framework to include empirical formulations of the breath cloud to incorporate the role of turbulent dispersion. Consequently, the fluctuating local velocity field and thermohygrometric conditions encountered by droplets within the exhaled plume can be approximated in a computationally efficient manner, evaluating the equations through time-stepped analytical solutions, although this results in lower accuracy regarding the flow field resolution compared to CFD analysis. Xie *et al.* [234] revisited the classical Wells evaporation-falling curve [235] by modelling the respiratory flow as a steady, non-isothermal jet. Their framework explicitly incorporates an empirical buoyancy model to correct the jet trajectory, assuming self-similar Gaussian profiles for velocity and temperature distributions to evaluate the local conditions surrounding the droplet. In a later study, Parienta *et al.* [236] utilised a steady, non-buoyant turbulent jet model derived from classical axisymmetric theory; while this approach assumed a hyperbolic decay of the centreline velocity, it simplified the fluid dynamics by neglecting thermal buoyancy effects. Redrow *et al.* [237] refined the aerodynamic representation by addressing the stochastic nature of the flow. They coupled the mean velocity field of a buoyant jet with a Discrete Random Walk (DRW) model to simulate the turbulent dispersion of droplets via interaction with instantaneous velocity fluctuations. More recently, Balachandar *et al.* [238] proposed a theoretical multiphase framework based on momentum conservation, segmenting the respiratory event into two distinct fluid dynamic stages: an initial momentum-dominated *jet phase* and a subsequent buoyancy-dominated *puff phase*. In this model, the spatial velocity within the cloud is approximated as uniform rather than Gaussian, simplifying the volume integration while capturing the entrainment dynamics during the jet-to-puff transition. Cavazzuti and Tartarini [239] proposed an analytical model of the breath cloud grounded in momentum conservation, analogous to the framework of Balachandar *et al.* [238]. However, their approach distinguishes itself by adopting a Gaussian profile for the jet velocity rather than a uniform distribution. Furthermore, this model incorporates a DRW model to explicitly simulate the turbulent dispersion of droplets.

This chapter presents a multiscale model [240] that couples the computational efficiency of analytical solutions for determining droplet transport and evaporation with the capability of CFD to resolve the complex, three-dimensional, and unsteady airflow patterns characteristic of breathing events. The application of multiscale models within this topic remains limited. D'Alessandro *et al.* [241] coupled an Eulerian-Lagrangian CFD simulation with a Population Balance Equation to explicitly model the nucleation of solid particles within the droplet, thereby refining the evaporation prediction. Mendez *et al.* [242] formalised a multiscale methodology designed to bridge CFD simulations at the microscale with population-scale epidemiological models. In their work, Eulerian-Lagrangian CFD simulations of a breathing individual were converted into concentration maps via a coarse-graining method; these maps

were subsequently employed to assess infection risk across various crowd scenarios. Mirzaei *et al.* [243] trained an Artificial Neural Network (ANN) using a dataset of 35 Eulerian-Lagrangian CFD simulations of a cough. This approach yielded a black-box model that accepts air velocity and thermohygrometric conditions as inputs and outputs the horizontal and vertical spread of released droplets. Finally, Chong *et al.* [244] integrated a dose-response model to generate spatial risk maps for indoor environments. By utilising data from Eulerian-Lagrangian CFD simulations of an environment containing an infected individual, their model calculates local infection risk, thus overcoming the limitations of the classical well-mixed assumption inherent in the widely used Wells-Riley model. Mazumdar *et al.* [245] developed a coupled CFD-analytical model for airliner cabins. The analytical part solves the 1D transport equation; however, since this approach fails to capture the complex contaminant distribution near the source, they used a CFD simulation in that region to provide the correct boundary conditions to the analytical model.

To overcome the inherent limitations of the analytical breath cloud model in capturing complex environmental flows, the multiscale model described in this work integrates CFD simulation of the exhaled airflow. However, unlike the cited approaches, which exploit full Eulerian-Lagrangian simulations, the CFD simulations used in this model are strictly limited to the continuous phase, *i.e.* humid air, and exclude the computational burden of tracking discrete particles. The analytical model instead utilises the CFD results as local boundary conditions, specifically air velocity, temperature, and humidity, to determine droplet transport and evaporation. This decoupling yields significant computational efficiency and flexibility: extensive statistical risk analyses can be conducted by varying droplet parameters (such as size, number, initial position, and ejection timing) without the need to repeat the computationally expensive airflow simulation.

5.3. Methodology

This chapter describes the numerical framework developed to simulate the dispersion of respiratory droplets. The proposed solution is a multi-scale model that couples an analytical Lagrangian solver [239] used to compute droplet trajectories and evaporation rates, with 3D Unsteady Reynolds-Averaged Navier-Stokes (U-RANS) CFD simulation of the respiratory events. The coupling strategy is one-way and sequential, meaning the CFD simulation is executed first to resolve the continuous airflow due to the respiratory event under investigation. Its results are then used as input boundary conditions for the analytical model, which determines the fate of the droplet until a certain stopping criteria is met. This architecture effectively bridges the different scales involved: the CFD simulation resolves the macroscopic airflow and its unsteady fluctuations, while the analytical model handles the microscopic physics of droplet transport and evaporation within that environment. To enable a statistical analysis, the analytical model includes randomisation elements. Droplets are released at

random positions across the mouth or nose and at random times during the exhalation, while turbulent dispersion is modeled using a discrete random walk approach. Finally, the results are post-processed by dividing the domain into a 2D grid of cells where droplet trajectories are aggregated to compute local viral concentrations. These data are processed through an inhalation model to generate viral inhalation maps, which are subsequently used to derive spatial risk maps for infection by direct inhalation and ground deposition. In this work, the model is applied to five respiratory events: mouth breathing, nose breathing, speaking, coughing, and sneezing.

5.3.1. Analytical model

The analytical model is briefly summarised here; for a detailed derivation and validation, the reader is referred to [239]. As previously stated, the analytical model governs the Lagrangian phase of the problem.

5.3.1.1. Transport

Droplets are hypothesised as perfect spheres moving under the effects of their weight, $\mathscr{W}$, and drag, $\mathscr{D}$:

$$\mathscr{W} = \frac{1}{6}\left(\rho_\mathrm{p} - \rho_\mathrm{m}\right)\pi \boldsymbol{g} D_\mathrm{p}^3 \qquad ; \qquad \mathscr{D} = \frac{1}{8} C_\mathrm{d} \rho_\mathrm{m} \pi \left\|\boldsymbol{u}_\mathrm{r}\right\| \boldsymbol{u}_\mathrm{r} D_\mathrm{p}^2 \,, \tag{5.1}$$

where ρ denotes the density, D_p represents the particle diameter, $\boldsymbol{g}$ is the gravitational acceleration, C_d denotes the drag coefficient, and $\boldsymbol{u}_\mathrm{r}{=}\boldsymbol{u}_\mathrm{p}{-}\boldsymbol{u}_\mathrm{m}$ is the relative velocity between the particle and the local humid air mixture. Applying Newton's second law of motion yields:

$$\dot{\boldsymbol{u}}_\mathrm{r} = \frac{\mathscr{W} + \mathscr{D}}{m_\mathrm{p}} \,, \tag{5.2}$$

where m_p is the mass of the particle. Integrating Equation (5.2) provides the particle relative velocity, and integrating a second time yields the position:

$$\boldsymbol{u}_\mathrm{r}(t) = \boldsymbol{u}_\mathrm{p}(t) - \boldsymbol{u}_\mathrm{m}(t) = \boldsymbol{u}_{\mathrm{r},0}\, \mathrm{e}^{-\mathrm{t}/\tau_\mathrm{d}} + \boldsymbol{u}_{\mathrm{r,t}} \frac{\tau_\mathrm{d}}{\tau_\mathrm{d,t}} \left(1 - \mathrm{e}^{-\mathrm{t}/\tau_\mathrm{d}}\right) \,, \tag{5.3}$$

$$\boldsymbol{s}_\mathrm{p}(t) = \boldsymbol{s}_{\mathrm{p},0} + \left(\boldsymbol{u}_\mathrm{m} + \boldsymbol{u}_{\mathrm{r,t}} \frac{\tau_\mathrm{d}}{\tau_\mathrm{d,t}}\right) t + \left(\boldsymbol{u}_{\mathrm{r},0} - \boldsymbol{u}_{\mathrm{r,t}} \frac{\tau_\mathrm{d}}{\tau_\mathrm{d,t}}\right) \tau_\mathrm{d} \left(1 - \mathrm{e}^{-\mathrm{t}/\tau_\mathrm{d}}\right) \,. \tag{5.4}$$

In Equations (5.3) and (5.4), τ_d is the particle characteristic drag time, and the subscripts "0" and "t" refer to the initial and terminal conditions, respectively. The relative terminal velocity is calculated as:

$$\boldsymbol{u}_{\mathrm{r,t}} = \sqrt{\frac{4(\rho_\mathrm{p} - \rho_\mathrm{m})\boldsymbol{g} D_\mathrm{p}}{3 C_{\mathrm{d,t}} \rho_\mathrm{m}}} \,, \tag{5.5}$$

and the characteristic drag time is:

$$\tau_{\mathrm{d}} = \frac{4\rho_{\mathrm{p}} D_{\mathrm{p}}}{3 C_{\mathrm{d}} \rho_{\mathrm{m}} \boldsymbol{u}_{\mathrm{r}}} \,. \tag{5.6}$$

By substituting the relative velocity, $\boldsymbol{u}_{\mathrm{r}}$, in Equation (5.6) with Equation (5.5), the formulation for the terminal drag characteristic time is obtained:

$$\tau_{\mathrm{d,t}} = \sqrt{\frac{4\rho_{\mathrm{p}}^2 D_{\mathrm{p}}}{3 C_{\mathrm{d,t}} \rho_{\mathrm{m}} (\rho_{\mathrm{p}} - \rho_{\mathrm{m}}) \boldsymbol{g}}} \,. \tag{5.7}$$

5.3.1.2. Evaporation

Starting from the convection-diffusion equation in stationary form, assuming no source or sink terms and neglecting the cross-diffusion of air in water, the variation of the diameter due to mass transfer from the droplet surface to the surrounding moist air is given by:

$$\dot{D}_{\mathrm{p}} = -\frac{2 D_{\mathrm{v,s}} \mathrm{Sh}}{\rho_{\mathrm{p}} D_{\mathrm{p}} R_{\mathrm{v}} T_{\mathrm{s}}} \frac{p}{p_{\mathrm{a,s}}} \left(p_{\mathrm{v,s}} - p_{\mathrm{v},\infty} \right) \,. \tag{5.8}$$

In Equation (5.8), the subscript "s" indicates properties at the droplet surface, subscript "a" denotes the air, subscript "v" refers to vapour, and the subscript "∞" indicates the local ambient conditions. Sh is the Sherwood number, which can be evaluated using the Ranz-Marshall empirical correlation [246, 247]; p is the pressure; T is the temperature; and R is the specific gas constant. Finally, D_{v} is the binary diffusion coefficient of water in air, which can be evaluated using the correlation proposed in [248]:

$$D_{\mathrm{v}} = 21.2 \cdot 10^{-6} \left[1 + 0.0071 \left(T_{\mathrm{s}} - 273.15 \right) \right] . \tag{5.9}$$

By introducing the evaporation rate K:

$$K = -\frac{d D_{\mathrm{p}}^2}{d t} = \frac{4 D_{\mathrm{v,s}} \mathrm{Sh}}{\rho_{\mathrm{p}} R_{\mathrm{v}} T_{\mathrm{a}}} \frac{p}{p_{\mathrm{a,s}}} \left(p_{\mathrm{v,s}} - p_{\mathrm{v},\infty} \right) \,, \tag{5.10}$$

Equation (5.8) becomes:

$$\dot{D}_{\mathrm{p}} = -\frac{K}{2 D_{\mathrm{p}}} \,. \tag{5.11}$$

Integrating Equation (5.11) yields the droplet diameter as a function of time:

$$D_{\mathrm{p}}(t) = \sqrt{D_{\mathrm{p,0}}^2 - K t} \,. \tag{5.12}$$

During the evaporation process, the evaporation rate K is not constant. However, it can be demonstrated that for small diameters, it tends towards a steady value which depends solely

on the particle density and the ambient thermohygrometric conditions.

5.3.1.3. Energy

The droplet's accumulated sensible thermal power is balanced by convective heating and evaporative cooling:

$$\dot{Q}_{sn} = \dot{Q}_{cv} + \dot{Q}_{ev} \,, \tag{5.13}$$

where

$$\dot{Q}_{sn} = m_{\mathrm{p}} c_{\mathrm{p,p}} \frac{dT_{\mathrm{p}}}{dt} = \frac{\rho_{\mathrm{p}} \pi D_{\mathrm{p}}^3}{6} c_{\mathrm{p,p}} \frac{dT_{\mathrm{s}}}{dt} \,, \tag{5.14}$$

$$\dot{Q}_{cv} = hS(T_\infty - T_{\mathrm{s}}) = \lambda_{\mathrm{s}} \pi D_{\mathrm{p}} \mathrm{Nu}(T_\infty - T_{\mathrm{s}}) \,, \tag{5.15}$$

$$\dot{Q}_{ev} = h_{\mathrm{lt}} \frac{dm_{\mathrm{p}}}{dt} = h_{\mathrm{lt}} \frac{\rho_{\mathrm{p}} \pi D_{\mathrm{p}}^2}{2} \frac{dD_{\mathrm{p}}}{dt} = -h_{\mathrm{lt}} \frac{\rho_{\mathrm{p}} \pi D_{\mathrm{p}}}{4} K \,. \tag{5.16}$$

In Equation (5.14), $c_{\mathrm{p,p}}$ is the specific heat of the particle; in Equation (5.15), λ is the thermal conductivity and Nu is the Nusselt number; and in Equation (5.16), h_{lt} is the latent heat of vaporisation. Therefore, Equation (5.13) can be rearranged as:

$$\frac{dT_{\mathrm{s}}}{dt} = \frac{6\lambda_{\mathrm{s}} \mathrm{Nu}(T_\infty - T_{\mathrm{s}})}{c_{\mathrm{p,p}} \rho_{\mathrm{p}} D_{\mathrm{p}}^2} - \frac{3h_{\mathrm{lt}} K}{2c_{\mathrm{p,p}} D_{\mathrm{p}}^2} \,. \tag{5.17}$$

Equation (5.17) can be integrated by neglecting the dependence of K on T_{s}, as the droplet temperature variation is small and limited in time. Thus, by hypothesising that the droplet temperature is uniform ($T_{\mathrm{p}} = T_{\mathrm{s}}$), the droplet temperature as a function of time is obtained:

$$T_{\mathrm{p}}(t) = T_{\mathrm{s},0}\, \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{e}}} + \mathrm{T_{s,t}} \left(1 - \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{e}}}\right) \,, \tag{5.18}$$

where $T_{\mathrm{s},0}$ is the initial droplet temperature. It should be noted that the solution obtained has a form similar to that of Equation (5.3).

The droplet characteristic time associated with evaporation is:

$$\tau_{\mathrm{e}} = \frac{c_{\mathrm{p,p}} \rho_{\mathrm{p}} D_{\mathrm{p}}^2}{6\lambda_{\mathrm{s}} \mathrm{Nu}} \,, \tag{5.19}$$

and the droplet terminal temperature (*i.e.*, the ambient wet-bulb temperature during evaporation) can be calculated as:

$$T_{\mathrm{s,t}} = T_\infty - \frac{h_{\mathrm{lt}} \rho_{\mathrm{p}} K}{4\lambda_{\mathrm{s}} \mathrm{Nu}} \,. \tag{5.20}$$

5.3.1.4. Non-volatile fraction

In the analytical model, the droplet is defined as a mixture of an evaporating aqueous liquid fraction and a non-volatile solid fraction (comprising salts, enzymes, mucus, and electrolytes). As the chemical composition evolves during the process, the mean particle properties change; these are defined as the weighted average of the individual component properties. For a two-component solution, where φ_{sl} is the solid volume fraction and $\varphi_{\mathrm{lq}} = 1 - \varphi_{\mathrm{sl}}$ is the liquid volume fraction, the density is expressed as:

$$\rho_{\mathrm{p}} = \rho_{\mathrm{sl}}\varphi_{\mathrm{sl}} + \rho_{\mathrm{lq}}(1-\varphi_{\mathrm{sl}})\,, \tag{5.21}$$

where ρ_{sl} and ρ_{lq} are the densities of the solid and liquid fractions, respectively. While the solid mass remains constant, the volume fraction varies as evaporation proceeds. Therefore, it is practical to evaluate the solid volume fraction as a function of the instantaneous diameter relative to the initial condition:

$$\varphi_{\mathrm{sl}}(t) = \varphi_{\mathrm{sl},0}\left(\frac{D_{\mathrm{p},0}}{D_{\mathrm{p}}(t)}\right)^3 . \tag{5.22}$$

The droplet specific heat is calculated as the mass-weighted average of the solid and liquid values, using the mass fraction ω:

$$c_{\mathrm{p,p}}(t) = c_{\mathrm{p,sl}}\omega_{\mathrm{sl}} + c_{\mathrm{p,lq}}(1-\omega_{\mathrm{sl}}) = \frac{\rho_{\mathrm{sl}}}{\rho_{\mathrm{p}}}\varphi_{\mathrm{sl}}c_{\mathrm{p,sl}} + \frac{\rho_{\mathrm{lq}}}{\rho_{\mathrm{p}}}(1-\varphi_{\mathrm{sl}})\,c_{\mathrm{p,lq}}\,. \tag{5.23}$$

The molar mass of the particle M_{p} is computed using the molar fraction y:

$$M_{\mathrm{p}} = M_{\mathrm{lq}}y_{\mathrm{lq}} + M_{\mathrm{sl}}(1-y_{\mathrm{lq}})\,, \tag{5.24}$$

where

$$y_{\mathrm{lq}} = \varphi_{\mathrm{lq}}\frac{M_{\mathrm{p}}}{M_{\mathrm{lq}}}\frac{\rho_{\mathrm{lq}}}{\rho_{\mathrm{p}}} = \left(1 + \frac{M_{\mathrm{lq}}}{M_{\mathrm{sl}}}\frac{\rho_{\mathrm{sl}}}{\rho_{\mathrm{lq}}}\frac{\varphi_{\mathrm{sl}}}{1-\varphi_{\mathrm{sl}}}\right)^{-1} . \tag{5.25}$$

Raoult's law [249] implies that the partial pressure of water vapour at the interface decreases proportionally to y_{lq}. Consequently, as the solvent evaporates and the solute phase becomes dominant, the evaporation rate diminishes. The evaporation process terminates when equilibrium is reached ($p_{\mathrm{v,s}}$=$p_{\mathrm{v},\infty}$), which corresponds to the condition y_{lq}=RH. At this point, the non-volatile volume fraction is defined as:

$$\varphi_{\mathrm{sl}} = \left(1 + \frac{M_{\mathrm{lq}}}{M_{\mathrm{sl}}}\frac{\rho_{\mathrm{sl}}}{\rho_{\mathrm{lq}}}\frac{RH}{1-RH}\right)^{-1} . \tag{5.26}$$

In this work, the initial non-volatile volume fraction is assumed to be 1% [234], with an average density of $1200\,\mathrm{kg\,m^{-3}}$, a specific heat of $1100\,\mathrm{J\,kg^{-1}K^{-1}}$, and a molar mass of $0.1\,\mathrm{kg\,mol^{-1}}$. By rearranging Equation (5.22) and assuming complete evaporation of the liquid phase such that only the solid fraction remains ($\varphi_{\mathrm{sl}}(t) = 1$), the solid core diameter is:

$$D_{\mathrm{sl}} = \varphi_{\mathrm{sl},0}^{1/3} D_{\mathrm{p},0}\,. \tag{5.27}$$

Applying Equation (5.27) yields a solid core diameter equal to 21.5% of the initial droplet diameter. When the equilibrium condition in Equation (5.26) is applied, the results indicate that evaporation ceases when the solid volume fraction reaches 82.2%, corresponding to a final droplet diameter of 23% of the initial size.

5.3.1.5. Turbulent dispersion and droplet randomisation

To mimic the interaction of droplets with turbulent eddies, a random walk model is implemented, as suggested by similar works in the literature [250, 251]. This is achieved by adding a velocity perturbation, $\boldsymbol{u}'$, to the local mean flow velocity, $\boldsymbol{u}_{\mathrm{m}}$. Under the assumption of isotropic turbulence, the velocity perturbation is defined as:

$$\boldsymbol{u}' = \boldsymbol{\eta}\sqrt{\frac{2}{3}k}\,, \tag{5.28}$$

where $\boldsymbol{\eta}$ is a normally distributed random vector and k is the turbulent kinetic energy. The value of this perturbation is updated at time intervals Δt_η, calculated as the minimum between the characteristic eddy lifetime and the time required for the droplet to cross the eddy:

$$\Delta t_\eta = \min\left[0.3\frac{k}{\varepsilon}\,,\, -\tau_{\mathrm{d}}\ln\left(1 - \frac{C_\mu^{3/4}\,k^{3/2}}{\tau_{\mathrm{d}}\,\|\boldsymbol{u}_{\mathrm{r}}\|\,\varepsilon}\right)\right], \tag{5.29}$$

where $C_\mu = 0.09$ and ε is the turbulent dissipation rate. The use of a random walk model allows for a statistical analysis of the droplet trajectories. To enhance this analysis, the initialisation of droplets is randomised in both space and time, meaning a droplet can be exhaled at any point across the mouth or nose surface and at any moment during the exhalation. These random initial values follow specific probability distributions: the position is based on the velocity profile, while the release time is based on the mass flow rate. As a result, droplets are most likely to be generated where and when the airflow is strongest. Finally, the initial velocity of the droplet is assumed to be aligned with the local airflow, with a magnitude set as a fraction of the local air velocity.

5.3.1.6. Inhalation model

To quantify the infection risk, an inhalation model is required. For a generic isolated respiratory event, consider a control volume V_c traversed by the released droplets. The *viral exposure* e_c [252] is defined as the viral concentration within the volume, weighted by the droplet residence time t_i:

$$e_c = \frac{1}{V_c} \sum_i Q_i t_i \,, \tag{5.30}$$

where Q_i is the viral load, expressed in *quanta* (*i.e., infectious doses*), carried by the i-th droplet. In the case of a periodic respiratory event, such as breathing, Equation (5.30) can be reformulated in terms of the mean viral concentration, c_c, within the volume by dividing the viral exposure by the event period:

$$c_c = \frac{e_c}{t_{pr}} = \frac{1}{V_c t_{pr}} \sum_i Q_i t_i \,. \tag{5.31}$$

Considering a susceptible host breathing normally, the number of *quanta*, i_c, inhaled from an infected volume V_{in} during a single breath is:

$$i_c = \frac{t_{in}}{t_{pr}} \sum_i Q_i \,, \tag{5.32}$$

where $t_{in}{=}t_{pr}{-}t_{ex}$ represents the inhalation time, and the ratio $\dfrac{t_{in}}{t_{pr}}$ represents the probability that the individual is inhaling while the droplet traverses the infected volume V_{in}. The summation in Equation (5.32) includes all droplets that satisfy a specific inhalation criterion. The inhalation volume V_{in} is approximated as a hemisphere located in front of the individual's mouth or nose, with a radius:

$$r'_{in} = \sqrt[3]{\frac{3V_{in}}{2\pi}} \,. \tag{5.33}$$

Assuming the inhaled flow is steady and directed radially inwards toward the mouth or nose, the magnitude of the inhalation velocity is a function of the radius r':

$$u_{in}\left(r'_{in}\right) = \frac{V_{in}}{2\pi t_{in} {r'_{in}}^2} \,. \tag{5.34}$$

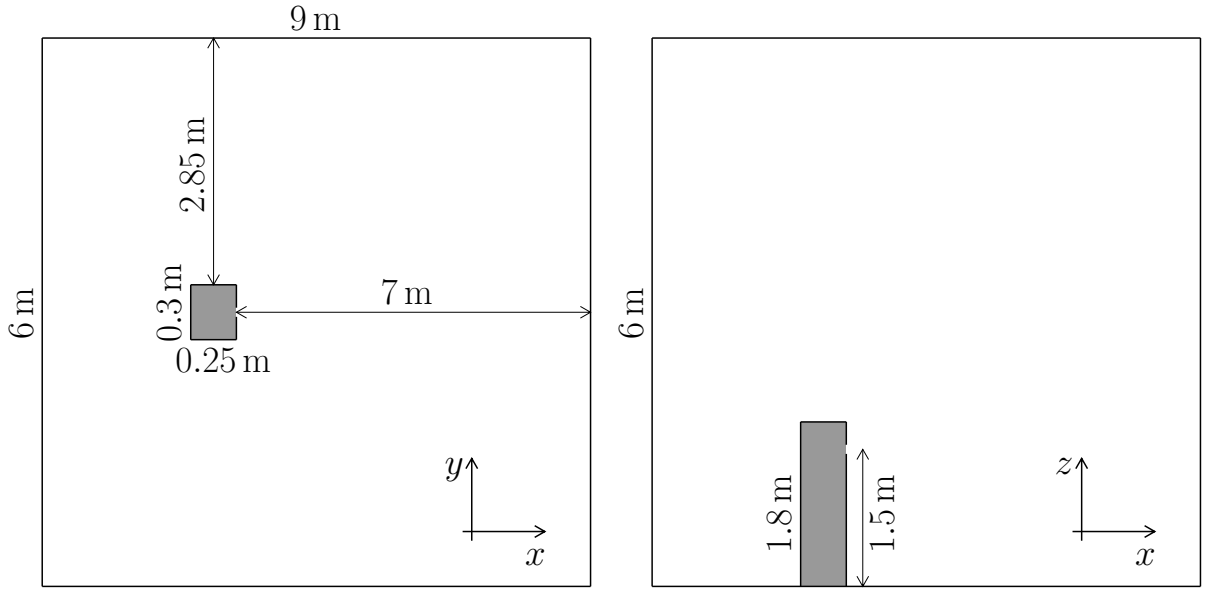
A droplet traversing V_{in} is considered inhaled (with a probability of t_{in}/t_{pr}) if its velocity component orthogonal to the radial direction is smaller in magnitude than the local inhalation velocity $\boldsymbol{u}_{in}$.

5.3.2. CFD model

The CFD model supplies the local environmental variables required by the analytical model, including turbulent kinetic energy and the turbulent dissipation rate. A series of U-RANS 3D CFD simulations are used to replicate the humid, warm airflow exhaled by a mannequin. Specifically, four different events are simulated: mouth breathing, nose breathing, coughing, and sneezing. This work also analyses the infection risk due to speaking, which has an exhalation similar to mouth breathing as it shares a comparable period and exhaled volume. The CFD model is implemented in OpenFOAM, using the custom solver *buoyantHumidityPimpleFoam*. This solver is based on the standard single-phase, pressure-based, compressible heat-transfer solver *buoyantPimpleFoam* but models the air as a humid mixture. No Lagrangian particles are injected into this simulation, as this task is handled by the analytical model. The $k-\omega$ SST turbulence model is adopted, as suggested in [228], with a turbulent Prandtl number set to 0.85. This turbulence model blends low and high Reynolds wall treatments; this feature is important in a domain where the friction Reynolds number varies significantly in space and time. In this study, the objective of the CFD is to model the unsteady and unbounded buoyant jet of the breath cloud; therefore, high solution accuracy at the boundaries is not required, except for the inlet surface. To ensure numerical stability, all advective terms are discretised using a first-order upwind scheme, except for the velocity divergence, which is discretised with a second-order upwind scheme. All diffusive terms are handled using a second-order central difference scheme, and time advancement is performed employing the first-order Implicit Euler scheme. The pressure-velocity coupling is solved using the Pressure-Implicit with Splitting of Operators (PISO) algorithm, with four inner coupling loops and one non-orthogonal corrector. The governing equations solved are the conservation of mass, momentum, energy, and species transport, where a unitary Schmidt number is assumed:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \,, \\ \dfrac{\partial (\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \rho \boldsymbol{g} + \nabla \cdot \left(\mu_{\text{eff}} \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right)\right) - \nabla \left(\dfrac{2}{3} \mu_{\text{eff}} \nabla \cdot \boldsymbol{u}\right) \,, \\ \dfrac{\partial (\rho (h + k'))}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} (h + k')) - \dfrac{\partial p}{\partial t} = \nabla \cdot (\rho \alpha_{\text{eff}} \nabla h) + \rho \boldsymbol{g} \cdot \boldsymbol{u} \,, \\ \dfrac{\partial (\rho Y)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} Y) = \nabla \cdot \left(\dfrac{\mu_{\text{eff}}}{\text{Sc}} \nabla Y\right) \end{cases} . \tag{5.35}$$

The simulation domain consists of a large, unventilated room where a mannequin is positioned far enough from the walls to prevent boundary effects. This allows the breath cloud to expand freely in front of the individual. The infected subject is modeled as a simple elongated hexahedron, following the approach of Zhang *et al.* [253]. As discussed in [254], a more detailed body shape would have minimal impact on the exhalation characteristics,



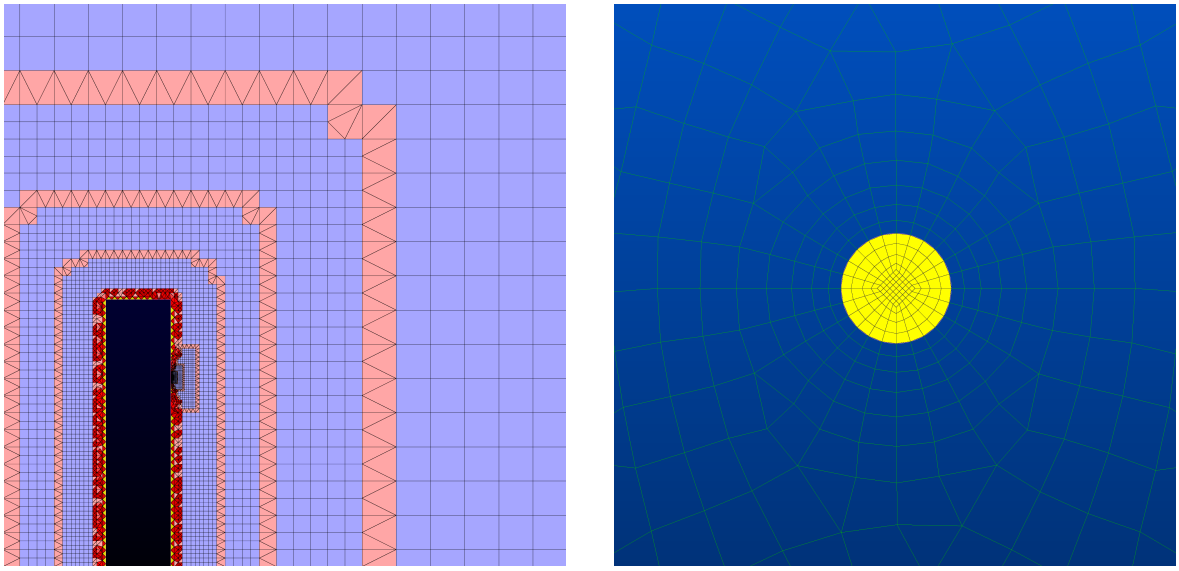
(a) Room top view **(b)** Room side view

Figure 5.1: CFD computational domain (dimensions are not scaled for clarity).

affecting only the local airflow around the body itself. This simplified geometry facilitates the use of a hexahedral-dominant grid, enabling finer cell size control and more accurate results at a reduced computational cost. The hybrid mesh is generated using Pointwise with a Voxel algorithm, which achieves high refinement near the mouth, where the steepest gradients are expected, while maintaining a coarser grid in the rest of the domain. The computational domain is schematised in Figure 5.1, while Figure 5.2 presents two views of the mesh: the first shows the cell cross-section on the x-z plane passing through the room center, and the second details the surface mesh near the mouth, which is modeled as a circular orifice. Table 5.1 summarises the boundary conditions. The no-slip condition is applied to all surfaces, and the room walls are adiabatic. To simulate the thermal plume rising from the body, which can influence the trajectory of smaller droplets, the mannequin surface is maintained at a fixed temperature representing the average clothing temperature, T_{cl}. According to the ASHRAE standard [255]:

$$T_{\mathrm{cl}} \approx T_{\mathrm{sk}} - J_{\mathrm{cl}}\, q''_{\mathrm{sk}}\,, \tag{5.36}$$

where the skin temperature is T_{sk}=36 °C, the heat flux from a standing individual at rest is q''_{sk}=70 W m^{-2} K^{-1}, and the insulation coefficient for a T-shirt is J_{cl}=0.72 clo (where 1 clo=0.155 m^2 K W^{-1}). At the inlet patch (mouth or nose), the temperature is fixed at 34 °C with a relative humidity of 95% [256]. Initially, the air in the domain is stagnant at 20 °C with 50% relative humidity. The velocity at the inlet patch is defined by a uniform, unsteady function. This function shares the same shape for all respiratory events, differing only in peak value and duration. Figure 5.3 illustrates this function, which is a piecewise sinusoidal curve:



(a) Volumetric mesh on the x-z plane **(b)** Surface mesh on the y-z plane

Figure 5.2: CFD mesh grading near the mannequin mouth.

Table 5.1: CFD simulation boundary conditions.

Boundary	Velocity	Temperature	Relative Humidity
Walls	No-slip	Adiabatic	Zero-gradient
Mannequin	No-slip	28 °C	Zero-gradient
Exhalation	Velocity function	34 °C	95%
Inhalation	Velocity function	Zero-gradient	Zero-gradient

the exhalation begins with a short acceleration phase reaching a peak at time t_{pk}, followed by a longer deceleration phase ending at t_{ex}. For periodic events, such as breathing, the inhalation is modeled using the same function with a reversed sign and period, repeating the exhalation-inhalation cycle. For non-periodic (isolated) events, inhalation is omitted. The direction of the exhaled flow varies by case; consequently, the orifice diameter is adjusted to match the target exhaled volume, V_{ex}, and peak velocity, u_{pk}, as summarised in Table 5.2. Since the velocity profile is sinusoidal, $u_{\mathrm{pk}}=\frac{\pi}{2}u_{\mathrm{avg}}$, and the inlet is modeled as a vertical circular surface, the mouth/nose diameter D_{mt} is calculated as:

$$D_{\mathrm{mt}} = \sqrt{\frac{2V_{\mathrm{ex}}}{u_{\mathrm{pk}} t_{\mathrm{ex}} \cos\theta}}\,, \tag{5.37}$$

where θ is the exhalation angle relative to the horizontal. The breathing period is set to 4 s, with exhalation occupying 60% of the cycle. For nose breathing, the inlet patch is split vertically into two halves to mimic the flow from nostrils, with the flow direction rotated

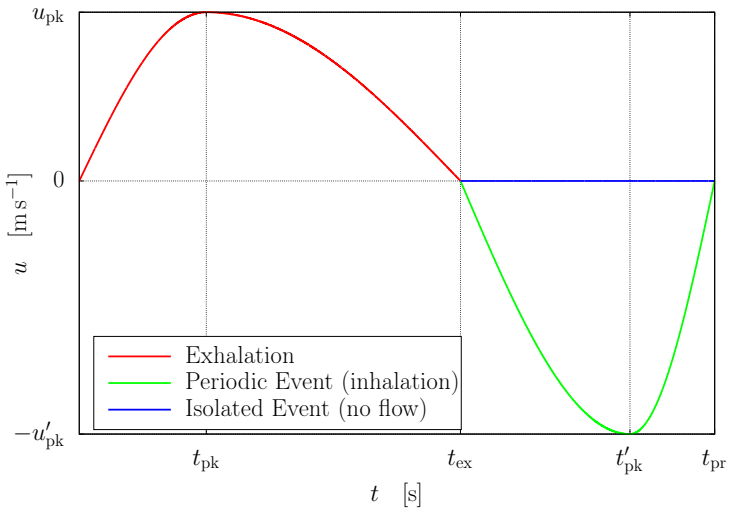


Figure 5.3: Inflow/outflow velocity boundary condition function imposed at the inlet.

Table 5.2: Modelling parameters for the mouth/nose boundary across different respiratory events.

Respiratory Event	Event Type	V_{ex} [dm^3]	u_{avg} [m s^{-1}]	u_{pk} [m s^{-1}]	t_{pk} [s]	t_{ex} [s]	t'_{pk} [s]	t_{pr} [s]	θ [°]	D_{mt} [mm]
Mouth Breathing	Periodic	0.6	1.27	2	0.8	2.4	3.5	4.0	−5.0	15.84
Nose breathing	Periodic	0.6	1.91	3	0.8	2.4	3.5	4.0	−60.0	18.26
Coughing	Isolated	1.0	6.37	10	0.1	0.4	–	∞	−27.5	23.74
Sneezing	Isolated	2.0	19.10	30	0.1	0.4	–	∞	+27.5	19.39

by ±10° relative to the vertical axis. The parameters in Table 5.2 aim to represent standard breathing, coughing, and sneezing events, although these characteristics naturally involve many variables and vary significantly between subjects. The data presented are derived from a critical review and averaging of literature values, noting that reported data often vary across studies. Peak and average velocities are based on [257, 258], directions and timing on [259, 260], and exhaled volumes on [261].

The simulation stops at 30 s, which is a sufficient time-span to permit to any droplet to settle to the ground or complete the evaporation and become an airborne particle completely carried by the local airflow. The time-step is adaptive, and in every iteration it is chosen to enforce that the maximum Courant number is < 1. For isolated events the results are saved every 0.02 s, while for periodic events every 0.1 s. This discrepancy is due to the fact that in the periodic events simulated, *i.e.* mouth and nose breathing, the velocity gradients are milder both in space and time. A mesh sensitivity analysis was conducted to assess the suitability of the computational domain and ensure that the results are independent of the spatial discretisation. Three hybrid grids with increasing levels of refinement were generated and compared by coupling the CFD simulation with the analytical model: a coarse grid with 250 000 elements and a minimum cell size of 0.6 mm near the mouth; an intermediate grid with 850 000 elements and a minimum cell size of 0.4 mm; a fine grid with 3.4×10^6 elements

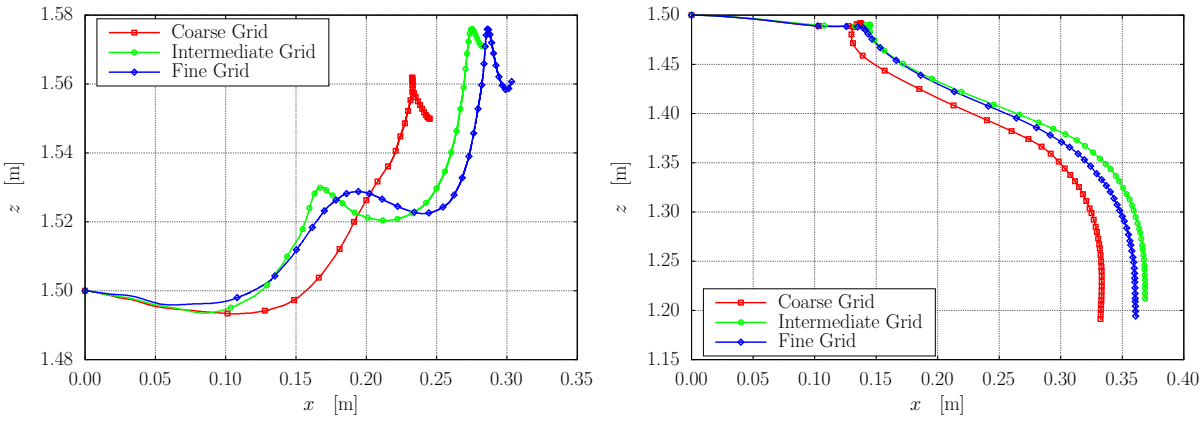
Table 5.3: Results of the grid sensitivity analysis: droplet trajectory normalised weighted-average distance with respect to the fine grid results for the two scenarios investigated.

Droplet Diameter	Mouth Breathing		Coughing	
	Coarse Grid	Intermediate Grid	Coarse Grid	Intermediate Grid
25 μm	18.35%	4.93%	5.69%	2.37%
40 μm	12.74%	8.24%	19.77%	11.95%
60 μm	14.31%	1.49%	28.97%	8.96%
90 μm	6.41%	0.28%	8.17%	4.48%
200 μm	–	–	0.30%	0.34%

and a minimum cell size of 0.1 mm. To assess the influence of the mesh resolution on droplet transport, the stochastic components of the analytical model, specifically turbulent dispersion and random initialisation, were disabled. This ensures that any observed deviation is solely attributable to the grid resolution. Droplets of various sizes were injected from the center of the mouth orifice just prior to the peak exhalation velocity. The trajectories computed on the coarse and intermediate grids were compared against those obtained from the fine grid, which served as the reference solution. The droplet positions were recorded at fixed time intervals, and the Euclidean distance from the reference position was calculated at each step. To quantify the discrepancy, a specific error metric was defined: the weighted average of these distances, normalised by the total length of the reference trajectory. The weighting factors increase linearly in time, such that the weight assigned to the final position is double that of the first. This rationale was chosen to attribute greater importance to the final portion of the trajectory, where cumulative errors due to grid differences are more likely to manifest. The analysis covered two distinct scenarios: mouth breathing (injection at $t = 0.6\,\text{s}$, tracking duration 4 s, time interval 0.1 s) and coughing (injection at $t = 0.08\,\text{s}$, tracking duration 1 s, time interval 0.02 s). Table 5.3 summarises the calculated error metrics, while Figure 5.4 visualises the trajectory comparison for two representative cases: a 25 μm droplet during breathing and a 90 μm droplet during coughing. It can be observed that the difference between grids becomes negligible for large droplets, as their motion is dominated by inertia and gravity rather than local air currents. Considering the consistently low error values, the intermediate grid was selected as the optimal compromise between numerical accuracy and computational cost.

5.3.3. Multi-scale coupling

The multi-scale model relies on a one-way coupling strategy between the Computational Fluid Dynamics (CFD) simulation and the analytical model. The unsteady CFD simulation of the exhaled airflow is executed first, and the resulting flow fields are saved at regular



(a) 25 µm droplet, mouth breathing scenario

(b) 90 µm droplet, coughing scenario

Figure 5.4: Results of the grid sensitivity analysis: droplet trajectories obtained from the different computational grids.

time intervals (0.1 s for periodic events and 0.02 s for isolated events). Upon completion of the CFD simulation, the fields required by the analytical model are interpolated at the grid points and stored in matrices, where the rows correspond to the grid points and the columns to the saved time-steps. The computational grid is then exported as a *vtkUnstructuredGrid* object to leverage the efficient query functions provided by the VTK library [262], facilitating the identification of element indices within the results matrices. Seven specific fields are required by the analytical model: the velocity components to compute droplet transport, temperature and relative humidity to evaluate the evaporation rate and energy balance, turbulent kinetic energy and turbulent dissipation rate for the turbulent dispersion model. The turbulent dissipation rate is derived from the turbulent kinetic energy and the specific dissipation rate as follows:

$$\varepsilon = C_\mu k \omega \, . \tag{5.38}$$

Due to the low droplet concentration characteristic of breathing events, droplet-to-droplet interaction is considered negligible [263]. Consequently, the analytical model simulates the droplets as an isolated batch. To accelerate query operations, information regarding the CFD grid and flow fields is loaded into memory only once at the beginning of the simulation. This process can be highly RAM-intensive depending on the problem size; for instance, loading data for a CFD simulation with 1×10^6 grid points and thousands of time-steps requires approximately 50 GB of RAM. To reduce computational costs and memory usage, only the subdomain where droplets are expected to remain is exported. Since the time-step of the analytical model typically differs from the CFD output frequency, and the droplet position does not necessarily coincide with a grid point, the seven fields of interest are interpolated in both space and time from the stored CFD result matrices. The field values at the specific droplet position and time are returned to the analytical model, which utilises them to compute

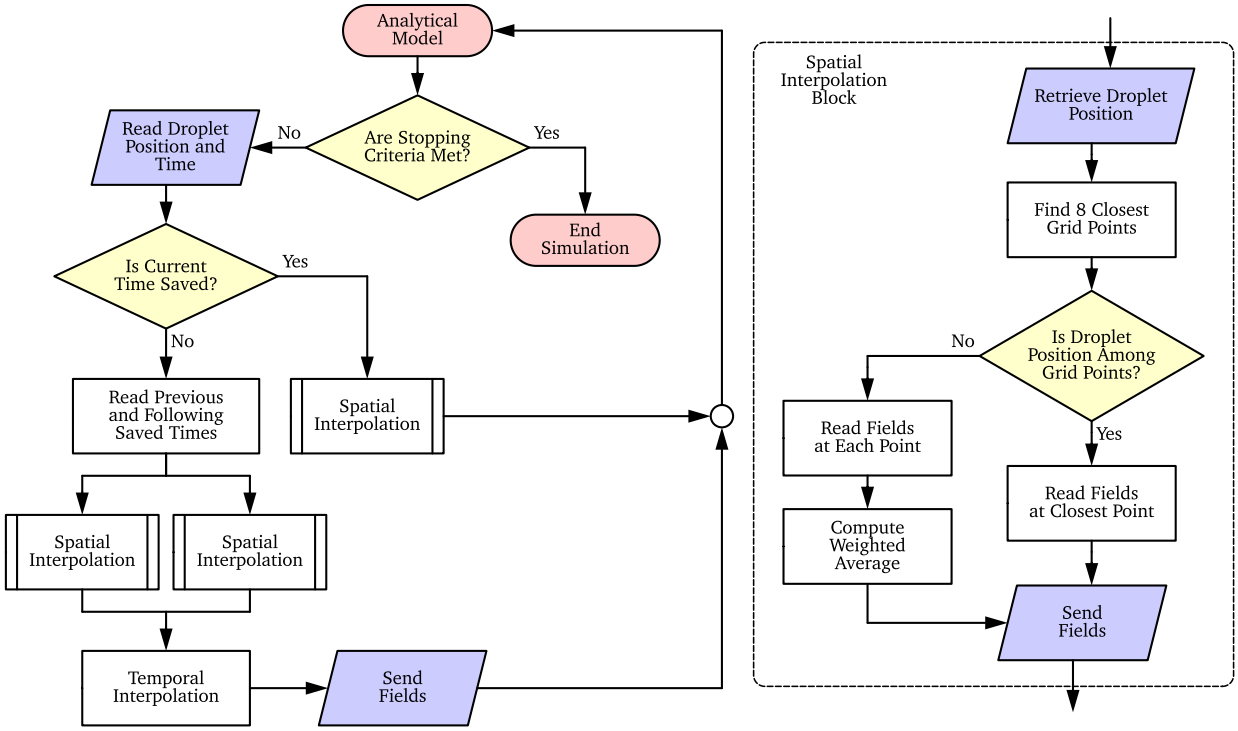


Figure 5.5: One-way multi-scale coupling flow chart.

the next position and diameter. Spatial interpolation is performed using the values from the eight grid points closest to the droplet. If the distance between the particle and the nearest grid point falls below a specific tolerance (10 nm in this case), the values at the closest point are used directly. Otherwise, an inverse distance weighted average is calculated based on the eight nearest neighbours. Temporal interpolation is linear; however, if the analytical solver time coincides exactly with a saved time-step, interpolation is bypassed. Figure 5.5 presents a flowchart detailing the coupled model algorithm. The simulation employs two stopping criteria: when the droplet settles on the ground or when it effectively becomes a passive airborne particle fully entrained by the air. As previously mentioned, the initial droplet position is randomly selected within the mouth or nostril inlet surface, and the release time is randomised across the exhalation duration. The initial droplet temperature and velocity are assumed to match the local CFD values (*i.e.*, the droplet is initially in thermal equilibrium and fully entrained by the airflow). However, a correction is applied to the initial velocity to account for the Gaussian velocity profile typical of jets. The velocity function shown in Figure 5.3 is scaled by a factor $e^{-\zeta^2}$, where ζ ranges from 0 at the centre of the inlet cross-section to 1 at the borders. Furthermore, the initial particle direction is set to form an angle of $\arctan(\beta\zeta)$ with the axial axis. This formulation accounts for the radial velocity component characteristic of a jet with an entrainment coefficient $\beta = 0.11$.

Table 5.4: Exhalation properties.

Respiratory Event	$D_{\min}$	$D_{\max}$	B	N_{D}	V_{D}	N_{V}	Q
Breathing	0.1 µm	10 µm	30 µm	$2.67 \times 10^{5}\,\mathrm{h}^{-1}$	$7.07 \times 10^{-4}\,\mathrm{mm}^{3}\,\mathrm{h}^{-1}$	2.33	2.31
Speaking	0.1 µm	100 µm	60 µm	$5.39 \times 10^{5}\,\mathrm{h}^{-1}$	$1.41 \times 10^{-1}\,\mathrm{mm}^{3}\,\mathrm{h}^{-1}$	$4.66 \times 10^{+2}$	$4.66 \times 10^{+1}$
Coughing	0.1 µm	1 mm	20 mm	2.00×10^{5}	$5.24\,\mathrm{mm}^{3}$	$1.73 \times 10^{+4}$	$1.73 \times 10^{+2}$
Sneezing	0.1 µm	1 mm	200 mm	2.00×10^{6}	$5.24 \times 10^{+1}\,\mathrm{mm}^{3}$	$1.73 \times 10^{+5}$	$1.73 \times 10^{+3}$

5.4. Exhalation data

The multi-scale model calculates the trajectories, velocities, and diameter evolution of the droplets. However, this information alone is insufficient to evaluate the infection risk. Consequently, additional hypotheses regarding the characteristics of the exhalation and the emitted viral load are required. These include the exhaled droplet size distribution and total count, the viral concentration within the saliva, and the infectious dose required to induce infection. Existing studies regarding the quantity and size distribution of exhaled droplets are often limited and inconsistent. This discrepancy may be attributed to the absence of measurement instrumentation capable of accurately resolving the wide range of scales characteristic of human exhalation. Consequently, published data appear to be heavily influenced by the sensitivity limits of the specific devices employed. Based on these observations, Bourouiba *et al.* [264] compared multiple datasets and concluded that the results are effectively approximated by a power-law Probability Density Function (PDF). Consistent with the analysis in [238], which further elaborates on this concept, a power-law PDF $\mathscr{B}$ with an exponent of 2 is adopted here as the most suitable approximation for all respiratory events:

$$\mathscr{B}\left(D_{\mathrm{p}}\right) = \frac{B}{D_{\mathrm{p}}^{2}}\,. \tag{5.39}$$

Equation (5.39) is applied over selected ranges of droplet diameters $[D_{\min}, D_{\max}]$ using different values of the coefficient B depending on the type of event. These values are extrapolated by fitting droplet size and concentration data from several works as revised in [264, 265] and are reported in Table 5.4. In the table, data regarding periodic respiratory events are expressed as rates per hour (assuming a period of 4 s, *i.e.*, 900 cycles per hour), while non-periodic events are reported per single occurrence. Literature indicates that smaller droplets are associated with a higher infection risk, primarily because they can penetrate deeper into the lungs [266]. However, since there is no quantitative data available to accurately model this effect, a theoretical assumption is required. In this work, the infectious dose $\mathscr{I}$ due to different droplet sizes, is approximated using a power-law PDF with power -1:

$$\mathscr{I}\left(D_{\mathrm{p}}\right) = I D_{\mathrm{p}}\,. \tag{5.40}$$

To calculate the number of droplets and the liquid volume exhaled in a generic range of droplet sizes $[D_{\min}, D_{\max}]$, it is possible to integrate over the diameter range:

$$N_{\mathrm{D}} = \int_{D_{\min}}^{D_{\max}} \frac{B}{D_{\mathrm{p}}^2}\, dD_{\mathrm{p}} = B\left(D_{\min}^{-1} - D_{\max}^{-1}\right) , \tag{5.41}$$

and

$$V_{\mathrm{D}} = \int_{D_{\min}}^{D_{\max}} \frac{B}{D_{\mathrm{p}}^2}\frac{\pi D_{\mathrm{p}}^3}{6}\, dD_{\mathrm{p}} = \frac{\pi B}{12}\left(D_{\max}^2 - D_{\min}^2\right) . \tag{5.42}$$

By assuming c as the viral concentration, the number of exhaled viruses is:

$$N_{\mathrm{v}} = cV_{\mathrm{D}} . \tag{5.43}$$

In this work, it is assumed that the viral concentration c in the saliva is uniform and its value is c=3.3 nl^{-1}, which corresponds to the average concentration for SARS-CoV-2 as suggested in [267]. The number of exhaled *quanta* is calculated as:

$$Q = \int_{D_{\min}}^{D_{\max}} \frac{B}{D_{\mathrm{p}}^2}\frac{\pi D_{\mathrm{p}}^3}{6}\frac{c}{ID_{\mathrm{p}}}\, dD_{\mathrm{p}} = \frac{\pi Bc}{6I}\left(D_{\max} - D_{\min}\right) , \tag{5.44}$$

where $\bar{I}$=N_{V}/Q is the average infectious dose, defined as the number of virus copies estimated to cause infection with a probability of 1−e^{-1}≈63% if inhaled. To calculate the average infectious dose, N_{v} and Q are evaluated over the full range of possible droplet diameters in respiratory events, [0.1 µm, 1 mm]. The value of $\bar{I}$ cannot be directly measured but can only be estimated with a certain margin of uncertainty from medical considerations. According to the literature, the average infectious dose for infectious diseases like influenza or SARS-CoV-2 is $\bar{I}$=100 [268]. Finally, the constant I is calculated as:

$$I = \frac{2\bar{I}}{D_{\max} + D_{\min}} , \tag{5.45}$$

and with $\bar{I}$=100, the constant is found to be I=2.00 × 10^5 m^{-1}.

5.5. Simulation campaign

The numerical investigation addresses five distinct respiratory scenarios performed within the previously described unventilated domain: mouth and nose breathing, speaking, coughing, and sneezing. To ensure statistical robustness, a dataset comprising over 40 000 individual droplet trajectories was generated for each scenario. The continuous diameter spectrum [0.1 µm, 1 mm] is discretised into 972 logarithmic distributed intervals. This ensures that the simulation resolution is consistent across several orders of magnitude. The bounds of

these intervals are defined as:

$$\left[10^{-(j+1)/243},\ 10^{-j/243}\right] \text{mm}\,, \quad \text{for } j = 0, \ldots, 971\,. \tag{5.46}$$

Within each interval, the representative diameter $D_{\text{p},j}$ is selected as the geometric mean of the bounds:

$$D_{\text{p},j} = 10^{-(j+0.5)/243}\,\text{mm}\,, \quad \text{for } j = 0, \ldots, 971\,. \tag{5.47}$$

To reconstruct the correct viral load distribution despite this artificial discretisation, a statistical weighting method is employed. First, the total exhaled *quanta* Q_j corresponding to each diameter range is calculated using Equation (5.44). Subsequently, an ensemble of $N = 42$ independent stochastic simulations is executed for each representative diameter $D_{\text{p},j}$. Each simulated droplet in this ensemble is assigned a specific "*quanta* load" $Q_i = Q_j/N$. This weighting technique ensures that the predicted viral load distribution remains accurate (consistent with Equation (5.39) and Table 5.4), regardless of the difference between the number of simulated trajectories and the actual physical droplet count. Any droplet sizes falling outside the specific physical range of a given respiratory event (Table 5.4) are simply discarded during the post-processing phase. Regarding the droplet composition, a non-volatile solid volume fraction of 1% is set as initial fraction. Consequently, following the logic explained in Section 5.3.1.4, the evaporation process is physically constrained to terminate when the droplet shrinks to a residual nucleus of approximately 22% of its initial diameter. The computational cost of the Lagrangian tracking is highly dependent on the particle inertia and the duration of the respiratory event; generally, smaller droplets and longer exhalations demand greater resources. Performance benchmarking on a cluster node equipped with Intel Xeon Gold 5320 processors (2.2 GHz) indicates an average execution time of 10–15 s per droplet. The overhead introduced by the multi-scale coupling increases the runtime by approximately 10–20% compared to the purely analytical model, exclusive of the time required for the preliminary CFD airflow solution. It is worth noting that, since the model treats droplets as isolated and independent entities, the entire simulation procedure is easily parallelisable.

5.6. Infection risk evaluation

After calculating the inhaled *quanta* i_{c} with Equation (5.32), the infection risk can be calculated as:

$$r_{\text{c}} = 1 - \text{e}^{-\text{i}_{\text{c}}}\,. \tag{5.48}$$

Equations (5.30)-(5.32) refer to the direct inhalation route, but the disease can also be transmitted by the airborne infection route, *i.e.* by inhaling aerosol particles that remain suspended in the environment or are transported by the local airflow even long after the

exhalation. To calculate the infection risk due to airborne particles, the number of exhaled *quanta* is partitioned into airborne and deposited contributions: $Q{=}Q_{\mathrm{ab}} + Q_{\mathrm{gr}}$. Assuming that the airborne contribution is uniformly mixed in the environment, the trend of airborne viral concentration in time can be modeled with the Wells-Riley equation [269]. For an indoor environment with volume equal to V_{rm}, the viral load concentration for an isolated and periodic event is:

$$c_{\mathrm{rm}}(t) = \frac{Q_{\mathrm{ab}}}{V_{\mathrm{rm}}} \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{rm}}} , \tag{5.49}$$

and

$$c_{\mathrm{rm}}(t) = \frac{\dot{Q}_{\mathrm{ab}}}{V_{\mathrm{rm}}} \tau_{\mathrm{rm}} \left(1 - \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{rm}}}\right) . \tag{5.50}$$

In the above equations τ_{rm} is the viral removal rate characteristic time, due to dilution with the ventilation air and the viral titer decay. Time integration of Equations (5.49) and (5.50) yields the time averaged concentrations:

$$\bar{c}_{\mathrm{rm}}(t) = \frac{Q_{\mathrm{ab}}}{V_{\mathrm{rm}}} \frac{\tau_{\mathrm{rm}}}{t} \left(1 - \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{rm}}}\right) , \tag{5.51}$$

and

$$\bar{c}_{\mathrm{rm}}(t) = \frac{\dot{Q}_{\mathrm{ab}}}{V_{\mathrm{rm}}} \tau_{\mathrm{rm}} \left[1 - \frac{\tau_{\mathrm{rm}}}{t} \left(1 - \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{rm}}}\right)\right] . \tag{5.52}$$

From Equation (5.51) it can be noted that, for isolated events, the maximum value of the viral load occurs at time $t = 0$, after which the value decreases with time. For periodic events, according to Equation (5.52), at time $t = 0$ the concentration is minimum and then grows with time, since the virus is emitted progressively, until reaching a steady state value. To estimate the risk of fomite transmission, *i.e.* infection resulting from contact with particles settled on surfaces, the previously derived equations are adapted by substituting the airborne *quanta* load Q_{ab} with the deposited fraction Q_{gr}, and replacing the room volume V_{rm} with the surface area of the specific surface considered.

The rate of viral titer decay varies significantly depending on the physical state of the droplet; specifically, the biological decay occurs more rapidly for aerosol particles suspended in the air compared to droplets settled on surfaces. Based on experimental measurements by van Doremalen *et al.* [270] and the considerations in the work by Cavazzuti and Tartarini [271], the characteristic decay time for airborne droplets is estimated as $\tau_{\mathrm{rm}}{=}45$ min, whereas for settled droplets it extends to $\tau_{\mathrm{gr}}{=}9$ h. Applying Equation (5.49) to an isolated event reveals a substantial disparity in viral persistence. While the airborne viral concentration decreases by one order of magnitude approximately 1 h 45 min after the end of the exhalation, achieving the same reduction for viral load on a surface requires approximately 21 h.

Regarding the inhalation infection risk, the direct inhalation is time-dependent and non uniform in space, whereas the indirect inhalation risk is time-varying but assumed to be

spatially homogeneous. By considering the case where a susceptible host is reached by the exhalation of an infected individual after an isolated respiratory event for a duration t, the cumulative viral exposure and total inhaled *quanta* are:

$$\hat{e}_{\mathrm{c}}(V_{\mathrm{c}}, t) = e_{\mathrm{c}}(V_{\mathrm{c}}) + \bar{c}_{\mathrm{rm}}(t)\, t \ , \tag{5.53}$$

and

$$\hat{i}_{\mathrm{c}}(V_{\mathrm{c}}, t) = i_{\mathrm{c}}(V_{\mathrm{c}}) + \bar{c}_{\mathrm{rm}}(t)\, V_{\mathrm{in}} \frac{t}{t_{\mathrm{pr}}} \ , \tag{5.54}$$

where the hat symbol indicates the superposition of the direct and indirect inhalation routes, t/t_{pr} represents the number of breaths during which the susceptible host is exposed to airborne virus, and $\bar{c}_{\mathrm{rm}}$ is the time-averaged concentration calculated via Equation (5.51). For periodic events, $\bar{c}_{\mathrm{rm}}$ must be evaluated using Equation (5.52), and the total viral exposure and inhaled *quanta* are:

$$\hat{c}_{\mathrm{c}}(V_{\mathrm{c}}, t) = c_{\mathrm{c}}(V_{\mathrm{c}}) + \bar{c}_{\mathrm{rm}}(t) \ , \tag{5.55}$$

and

$$\hat{i}_{\mathrm{c}}(V_{\mathrm{c}}, t) = i_{\mathrm{c}}(V_{\mathrm{c}}) \frac{t}{t_{\mathrm{pr}}} + \bar{c}_{\mathrm{rm}} V_{\mathrm{in}} \frac{t}{t_{\mathrm{pr}}} \ . \tag{5.56}$$

Note that while Equation (5.48) considered only the inhaled *quanta* due to direct inhalation, to account also for the airborne infection risk, the value i_{c} must be substituted with the cumulative dose $\hat{i}_{\mathrm{c}}$. The viral concentration on a contaminated surface S_{c}, at a time t following an isolated or periodic event, varies both in space and time and is given by:

$$c_{\mathrm{c}}(S_{\mathrm{c}}, t) = \frac{\sum_i Q_i}{S_{\mathrm{c}}} \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{gr}}} \ , \tag{5.57}$$

and

$$c_{\mathrm{c}}(S_{\mathrm{c}}, t) = \frac{\sum_i Q_i}{S_{\mathrm{c}} t_{\mathrm{pr}}} \tau_{\mathrm{gr}} \left(1 - \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{gr}}}\right) \ . \tag{5.58}$$

The summation in Equations (5.57) and (5.58) accounts for all particles deposited on the surface S_{c}. If the deposited viruses are contacted and fully inhaled, the effective *quanta* dose is:

$$i_{\mathrm{c}} = c_{\mathrm{c}} S_{\mathrm{c}} \ . \tag{5.59}$$

In the results presented in this thesis, the room volume corresponds to the domain simulated with CFD (V_{rm}=324 m^3), and all quantities resulting from Equations (5.53)-(5.58) are calculated assuming an exposure time of 1 h, starting from a virus-free environment at $t = 0$. To obtain spatial risk maps, the volume next to the individual and the ground are discretised into cubic/square cells of 4 mm in size. For each cell, the infection risk equations are solved based on the trajectories derived from the multi-scale model. The inhalation

model detailed in Section 5.3.1.6, necessary to quantify the inhaled quanta, is applied to each cell by considering the inhaled volume centred at the cell centre. This process yields spatial risk maps of viral exposure, inhaled *quanta*, and infection probability. To improve the visualisation, the three-dimensional maps are projected onto the x-z plane, as this is the primary plane of trajectory evolution, and the concentration along the spanwise direction y is averaged based on the estimated breath cloud width and orientation.

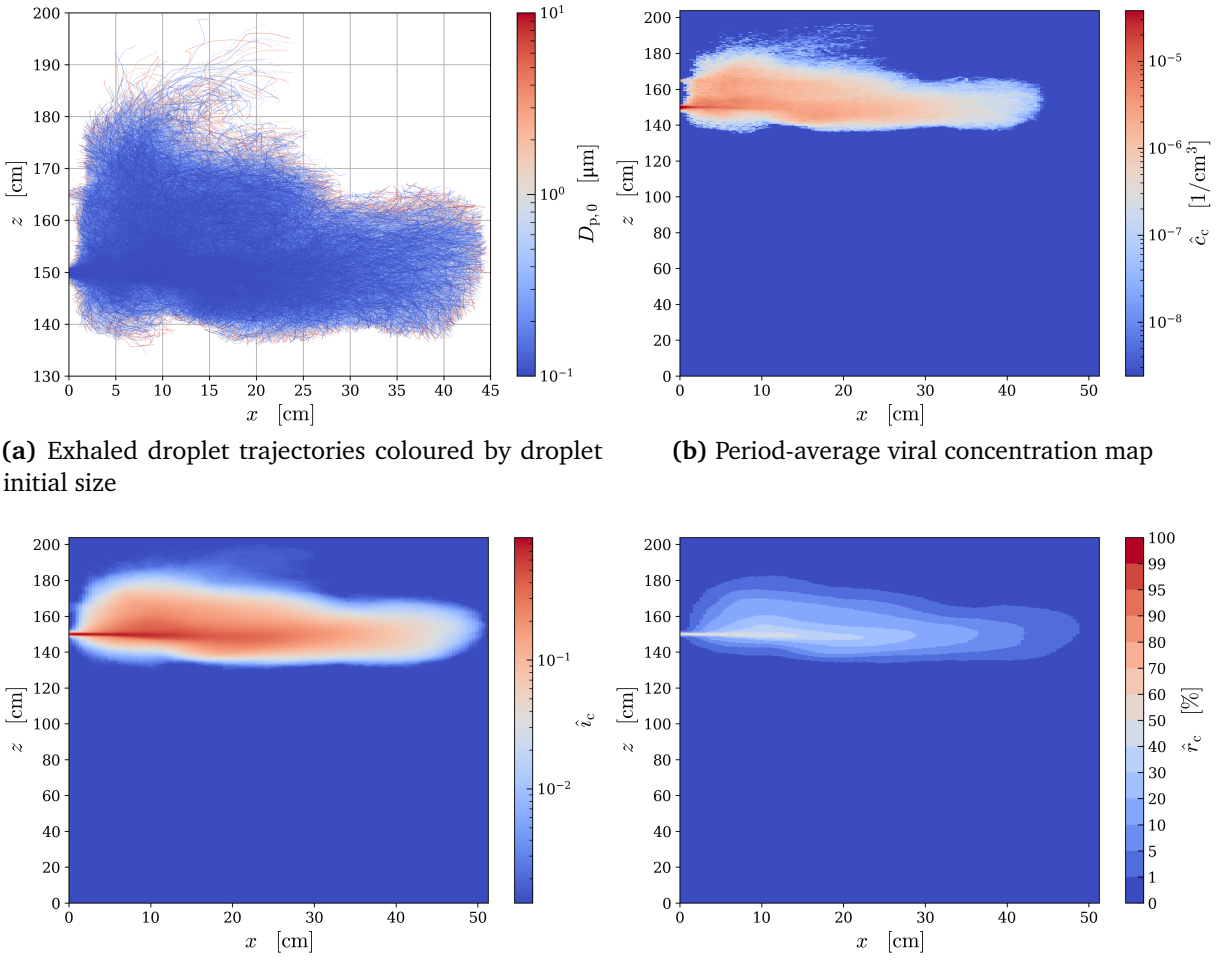
5.7. Results

As previously mentioned, the multi-scale model was employed to investigate five distinct scenarios: mouth breathing, nose breathing, speaking, coughing, and sneezing. The results are presented in terms of viral exposure, viral concentration, inhaled *quanta*, and infection risk maps. As specified in Section 5.4, the results refer to the SARS-CoV-2 virus; however, the simulation data can be post-processed by considering viral concentrations characteristic of any arbitrary infectious disease. Table 5.5 summarizes the airborne and deposited fractions of the exhaled *quanta* derived from the simulations for each case. In the breathing scenario, as shown in Table 5.4, the droplet diameter range is limited to a maximum of 10 µm. Consequently, there are no droplets with sufficient volume for gravitational forces to dominate over drag, resulting in no deposition. Conversely, for isolated events, the diameter of exhaled droplets spans the full range [0.1 µm, 1 mm]. In these cases, the majority of exhaled *quanta* settle on the ground, while the airborne fraction remains at approximately 10%. This behavior results from the *quanta* distribution assumed in Equation (5.44) and the specific value of the critical diameter D_{cr}. The critical diameter is defined as the initial size threshold below which droplets remain airborne and above which they settle. D_{cr} should not be interpreted as a single sharp cutoff, but rather as a transition range, since the fate of a droplet depends not only on its initial size but also on its specific trajectory history. For non-periodic events,

Table 5.5: Airborne and deposited *quanta* for each type of respiratory event.

Respiratory Event	Q_{ab}	Q_{gr}	D_{cr}[b]	$\bar{c}_{rm}$[a]	$\hat{\imath}$–i_c[a]	S_{gr}	c_{gr}[a]
Breathing	$2.31\,h^{-1}$	—	—	$2.39\times10^{-3}\,m^{-3}$[d]	1.29×10^{-3}[f]	—	—
Speaking	$44.33\,h^{-1}$	$2.28\,h^{-1}$	95.1 µm	$4.59\times10^{-2}\,m^{-3}$[d]	2.48×10^{-2}[f]	$88.5\,cm^2$	$2.44\times10^{-2}\,cm^{-2}$[h]
Coughing	15.62	157.17	90.5 µm	$2.66\times10^{-2}\,m^{-3}$[c]	1.44×10^{-2}[e]	$0.546\,m^2$	$2.58\times10^{-2}\,cm^{-2}$[g]
Sneezing	186.79	1541.09	108.2 µm	$3.18\times10^{-1}\,m^{-3}$[c]	1.72×10^{-1}[e]	$6.067\,m^2$	$2.27\times10^{-2}\,cm^{-2}$[g]

[a] Quantity evaluated at t=1 h
[b] From Eq. (5.44)
[c] From Eq. (5.51)
[d] From Eq. (5.52)
[e] From Eq. (5.54)
[f] From Eq. (5.56)
[g] From Eq. (5.57)
[h] From Eq. (5.58)



(a) Exhaled droplet trajectories coloured by droplet initial size

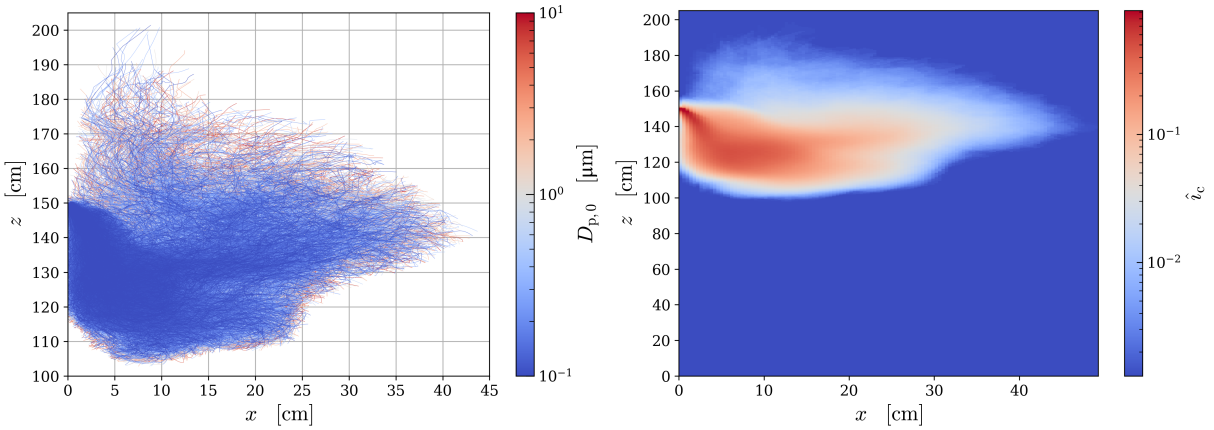
(b) Period-average viral concentration map

(c) Viral inhalation map after one hour exposure

(d) Risk of infection map after one hour exposure

Figure 5.6: Results of the mouth breathing scenario.

under the selected thermohygrometric conditions, the critical diameter is found to be around D_{cr}=100 µm. Using the airborne *quanta* fraction and the room volume, the time-averaged airborne viral concentration $\bar{c}_{rm}(t)$ is derived via Equations (5.51) and (5.52). The inhaled infectious dose is calculated using Equations (5.54)–(5.56). The analysis of the fomite transmission route is based on the spatially averaged ground viral concentration c_{gr} over the reference time-span, evaluated via Equations (5.57) and (5.58). The infected surface area S_{gr} and the *quanta* load settled on the ground are estimated directly from the multi-scale model results. As expected, the time-averaged airborne viral concentration in the room increases with the airborne *quanta* exhaled during the event. In contrast, the average ground concentration at t=1 h remains similar for speaking, coughing, and sneezing scenarios. In fact, although the amount of deposited *quanta* varies significantly, the infected surface area also changes in a nearly proportional way.



(a) Exhaled droplet trajectories coloured by droplet initial size

(b) Viral inhalation map after one hour exposure

Figure 5.7: Results of the nose breathing scenario.

5.7.1. Mouth breathing

Results for the mouth breathing scenario are presented in Figure 5.6. As previously mentioned, all droplets exhaled during breathing remain airborne, as their size is insufficient for gravitational forces to overcome drag. Following exhalation, they disperse within the breath cloud, driven by turbulent eddies and the upward lifting action of the body's thermal plume, as depicted in Figure 5.6a, where smaller droplet trajectories are plotted over larger ones. From the figure, it is evident that in this scenario, droplets linger in front of the individual up to a distance of ≈ 0.5 m. It is important to recall that no ventilation is present in this case and the trajectories of these small droplets are highly dependent on the local air streams. The spatial map in Figure 5.6b shows the time-averaged viral concentration resulting from both direct inhalation and the airborne fraction; this map is obtained by solving Equation (5.55) for each cell of the map. Figure 5.6c displays the viral inhalation map after one hour of exposure, obtained by solving Equation (5.56). In this scenario, the *quanta* emission rate is low; therefore, viral inhalation remains contained, although it reaches values close to unity near the emitter, showing a peak within 15 cm from the mouth. At distances greater than 50 cm from the mouth, the direct inhalation route is null, leaving only the inhalation of airborne particles, which corresponds to the lower scale of the colourbar. By applying Equation (5.48), the risk map reported in Figure 5.6d is obtained. This is a direct consequence of the values shown in Figure 5.6c; in fact, outside the breath cloud, the risk of infection is negligible, showing the same distribution of the inhalation map.

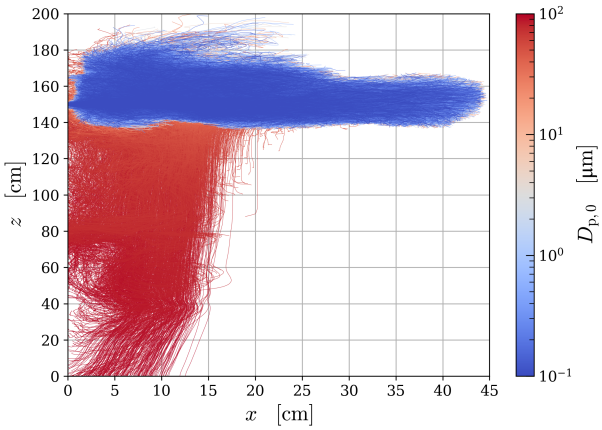
5.7.2. Nose breathing

The nose breathing scenario shares the same fundamental aspects as mouth breathing; however, the velocity and orientation of the emitted jet differ, as reported in Table 5.2. Consequently, the horizontal extension of the trajectories is reduced, causing the droplets to

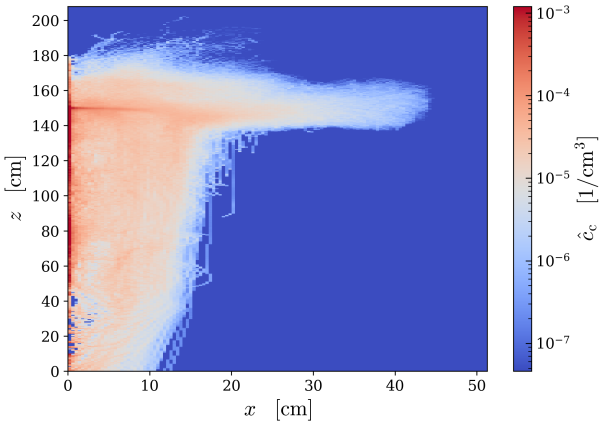
remain in closer proximity to the infected individual, as illustrated in Figure 5.7a. In this configuration, the upward deflection caused by the body's thermal plume becomes more evident. Simultaneously, the Coandă effect interacts with the airflow near the body surface, deviating the closest droplets downward. To maintain conciseness, only the trajectories and viral inhalation map in Figure 5.7b are presented for this scenario. The viral inhalation map demonstrates that, compared to the mouth breathing, the region of high viral intake is shifted nearer to the subject and positioned below the mouth level.

5.7.3. Speaking

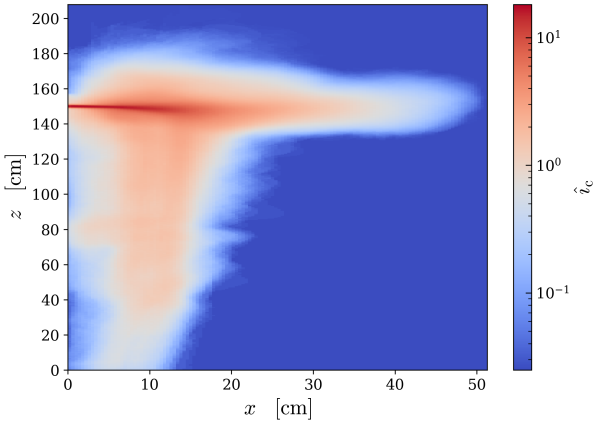
The speaking scenario shares similar exhalation properties to mouth breathing. As shown in Table 5.4, the only differences are the doubled coefficient B and the wider diameter range. Since these quantities affect only the post-process phase, the speaking scenario relies on the same simulation used for mouth breathing. For this reason, the trajectories for droplets up to 10 µm in size are identical to the ones in the mouth breathing scenario. As shown in Figure 5.8a, droplets with an initial diameter ≥ 40 µm are able to leave the breath cloud under the action of gravity and eventually fall to the ground. However, most of the droplets exiting the breath cloud become airborne. In fact, from Table 5.5, the critical diameter in this case is 95 µm; this value is very close to the maximum initial diameter reported in Table 5.4 (100 µm), confirming that most of the released droplets do not reach the ground. The trajectories show that larger droplets leave the area of influence of the breath cloud sooner, falling mostly vertically due to drag damping. For this reason, Figure 5.8b shows a high viral concentration close to the individual. The infection risk (Figure 5.8d) is much higher compared to the breathing scenario; this is due to the fact that the *quanta* emission rate during speaking is approximately twenty times the rate during breathing. High risk values occur not only inside the breath cloud but also in the region of falling droplets, at a distance of 10-20 cm from the infected individual. Keeping a longer distance from the infected individual is not a problem in environments such as offices or classrooms, but in environments with a relatively small volume, like a car cabin, this is not always possible. For long exposure times, airborne infection is likely to occur; in fact, according to Table 5.5, the virus inhaled after just 1 h of exposure to an infected speaking individual is 0.025 *quanta*. The droplets deposited on the ground are quite dispersed, as shown in Figure 5.8f, covering a limited area of $\approx 12 \times 8\,\mathrm{cm}^2$. The ground viral concentration map is generated by solving Equation (5.58) for every cell of the map. To ensure the result reflects the physical symmetry along the x-axis, the values from the opposite half-planes are averaged. Finally, the map is smoothed by averaging the value of each cell with its surrounding neighbours.



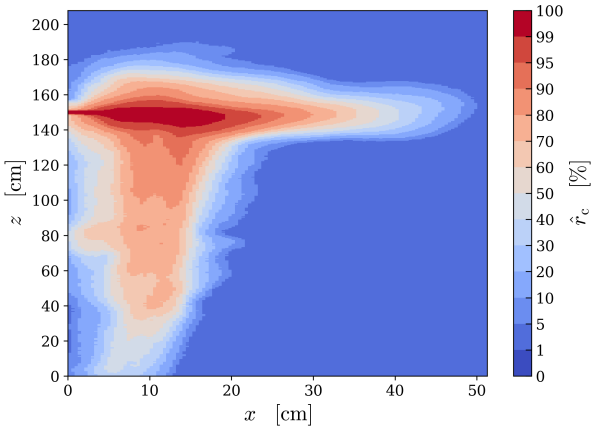
(a) Exhaled droplet trajectories coloured by droplet initial size



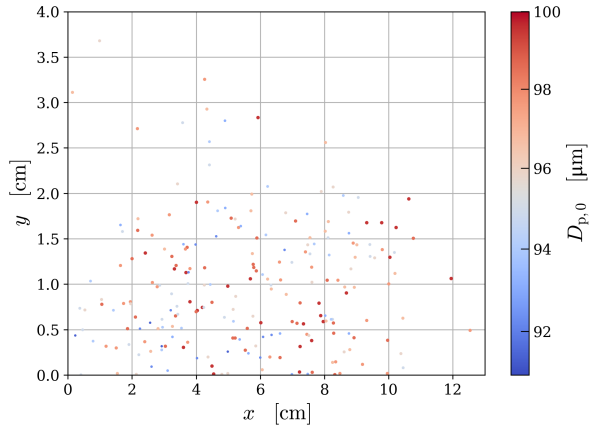
(b) Period-average viral concentration map



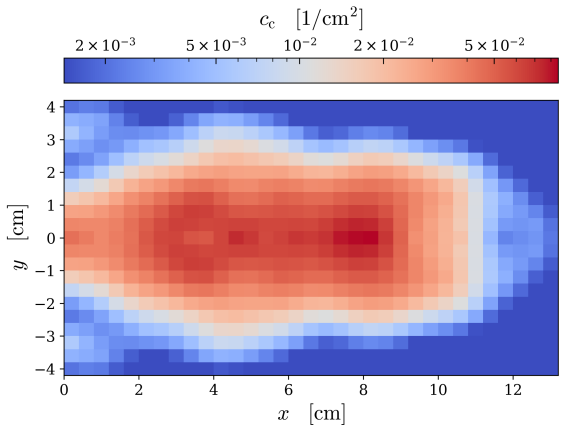
(c) Viral inhalation map after one hour exposure



(d) Risk of infection map after one hour exposure

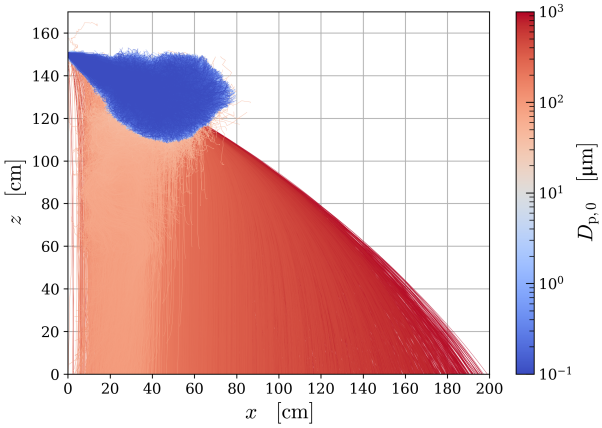


(e) Exhaled droplet ground deposition coloured by droplet initial size

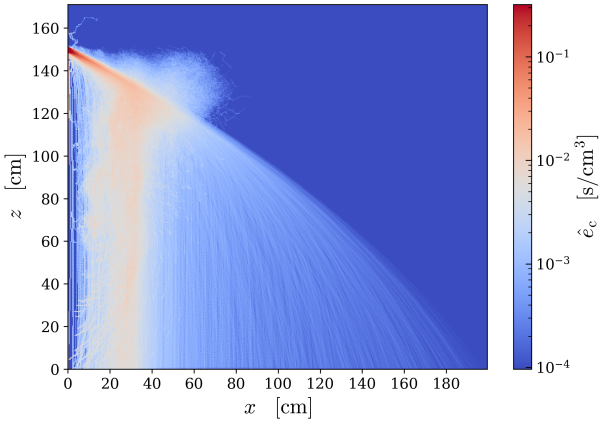


(f) Viral concentration map at the ground after one hour exposure

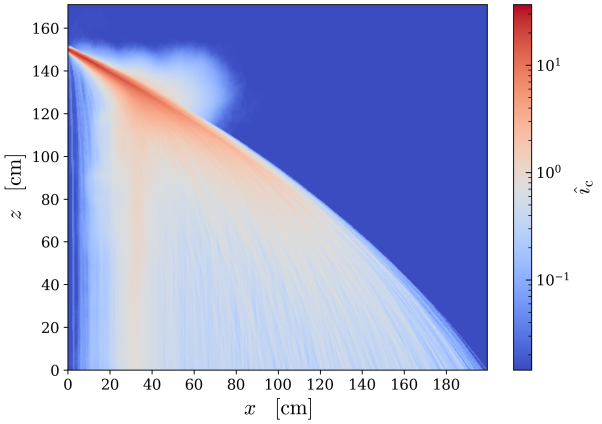
Figure 5.8: Results of the speaking scenario.



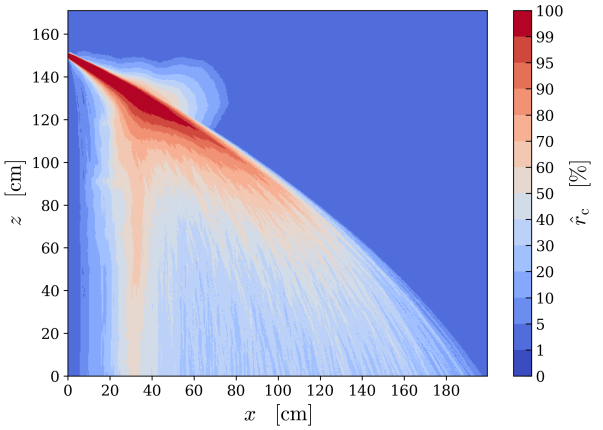
(a) Exhaled droplet trajectories coloured by droplet initial size



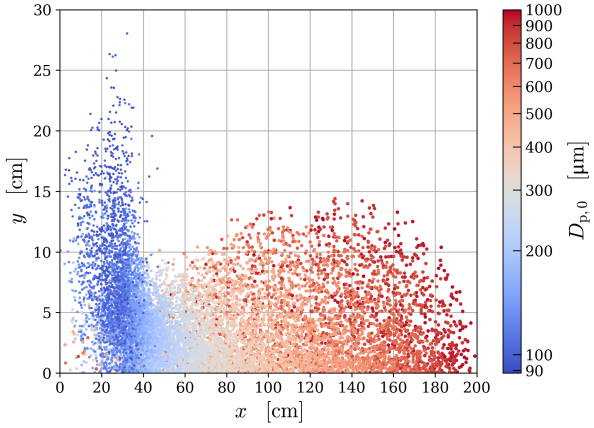
(b) Viral exposure map



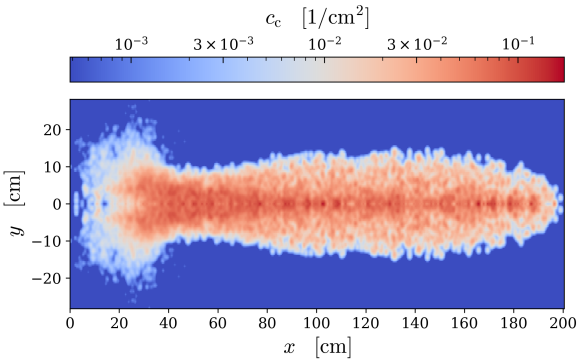
(c) Viral inhalation map after one hour exposure



(d) Risk of infection map after one hour exposure



(e) Exhaled droplet ground deposition coloured by droplet initial size



(f) Viral concentration at the ground after one hour exposure

Figure 5.9: Results of the coughing scenario.

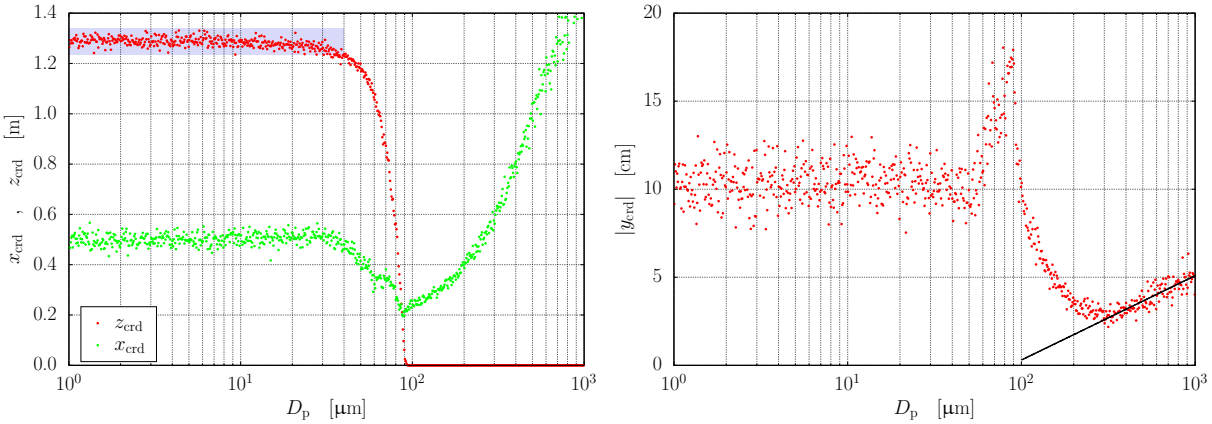
5.7.4. Coughing

The coughing scenario is characterised by a significantly higher quantity of droplets (see Table 5.4) and a higher exhalation velocity (see Table 5.2) compared to the previous cases. The results for the coughing scenario are presented in Figure 5.9. The critical diameter, 90 μm in this case, is slightly smaller than the one observed in the speaking scenario; this is partly attributed to the downward orientation of the exhalation (see Table 5.2), which favours droplet deposition. An analysis of Table 5.5 reveals that the total exhaled *quanta* from a single cough is almost four times greater than the load accumulated during 1 h of continuous speaking. This is primarily due to the presence of larger droplets (the maximum diameter for coughing is 1 mm), which settle to the ground. In contrast, the *quanta* associated with airborne particles are equivalent to the amount exhaled during approximately 20 minutes of speaking. From Figure 5.9a, it is possible to distinguish different trajectory behaviours depending on droplet size. Larger droplets fall to the ground following trajectories that become increasingly parabolic as the size grows; this occurs because inertial forces dominate over transport by the local airflow. Conversely, smaller droplets are fully transported by the breath cloud, becoming airborne once the cloud disperses into the environment. Finally, intermediate-sized droplets can exhibit both behaviours; for these particles, neither force totally dominates the other, and the fate of the droplet depends on its individual history. Most intermediate droplets leave the cloud under gravity, but due to evaporation, their size may become small enough to be eventually fully transported by the local airflow.

High exhalation velocities result in deep cloud penetration into the environment, reaching a distance almost double that of the speaking scenario. As reported in Figure 5.9b, viral exposure (calculated via Equation (5.53)) remains high up to 30 cm from the individual, primarily due to intermediate-sized droplets exiting the breath cloud. The concentration associated with the trajectories of larger droplets is lower, even though they carry a higher viral load; this occurs because these droplets disperse over larger volumes, extending up to 2 m from the individual, which dilutes the local concentration. From the figure, it is possible to observe a sudden deflection of intermediate droplets as they leave the cloud; this is caused by shear layer vortices generated at the interface between the cloud boundaries and the surrounding air. The viral inhalation map in Figure 5.9c is generated using Equation (5.54); subsequently, the risk map in Figure 5.9d is derived by applying Equation (5.48). All results for the coughing scenario consider a single event. Therefore, if multiple coughs were considered, the maps in Figures 5.9b and 5.9c would scale linearly with the number of coughs, while the infection risk map (Figure 5.9d) would scale according to the trend of Equation (5.48). According to Table 5.5, the amount of airborne *quanta* in this scenario is lower than that accumulated during 1 h of speaking; consequently, the risk associated with the airborne route is also lower. The infection risk due to airborne particles is not particularly high; however, as in the speaking case, it could become significant given longer exposure

times or multiple coughs by the infected individual in a closed environment. Regarding virus inhalability, larger droplets pose a greater threat via direct inhalation, causing a high risk of infection up to 1.5 m from the coughing individual. Thus, a social distance of 1 m is insufficient in this scenario to avoid direct inhalation. Figure 5.9e shows the droplet deposition on the ground, where each droplet is coloured according to its initial diameter. Larger droplets fall to the ground, outlining a nearly conical surface that closes smoothly near the maximum horizontal distance reached. On the other hand, smaller droplets that settle on the ground are more affected by the local airflow, showing greater dispersion in the y direction. Using the droplet deposition data and applying Equation (5.57), the viral concentration map in Figure 5.9f is obtained. The data in the figure are made symmetrical as previously discussed for the speaking scenario. The viral concentration on the ground forms a *fish-shaped* pattern, with the concentration decreasing with horizontal distance and showing evident dilution in the region where small droplets are dispersed.

In the coughing scenario, the full range of droplet sizes [0.1 μm, 1 mm] is expelled; therefore, in this case, a more complete analysis of the different droplet behaviours can be performed. From the results in Figure 5.9, a qualitative distinction between small, intermediate, and large droplets is made; however, analysing the droplet positions at the end of the simulation can provide a more quantitative distinction. The distinction between small and intermediate droplets can be identified by looking at the final z coordinate, since small droplets are fully transported by the breath cloud, while intermediate droplets leave the cloud due to gravity, reaching a lower z coordinate. To distinguish between intermediate and large droplets, it is more suitable to look at the final y coordinate. Since both types of droplets fall to the ground, the distinction relies on inertia: large droplets are less deviated in the y direction. It is worth noting that any deviation in the y direction can be due to two different effects: transport by the local airflow and the droplet initial velocity component in the y direction. The first effect influences small droplets more, while the latter affects large droplets more. Figure 5.10a displays the average final coordinates reached by the droplets at the end of the simulation. Since the coordinates are averaged over droplets of the same initial size, each dot represents the mean of 42 independent simulations of the same diameter, for a total of 972 data points. The final x and z coordinates are plotted in Figure 5.10a, where the shaded region in the top left denotes the average z coordinate of droplets up to 40 ± 2.58 μm. For droplets larger than 40 μm, the effect of gravity starts to dominate, causing the final z coordinate to decrease. Similarly, the horizontal distance (x coordinate) also decreases for droplets larger than 40 μm, reaching a minimum at the critical diameter; beyond this value, the trend is reversed. Figure 5.10b reports the final absolute y coordinate, showing a wider dispersion in the y direction for small droplets and a peak corresponding to the critical diameter. For intermediate droplets with a sub-critical initial diameter, dispersion increases proportionally with droplet size; this is attributed to



(a) x and z coordinates

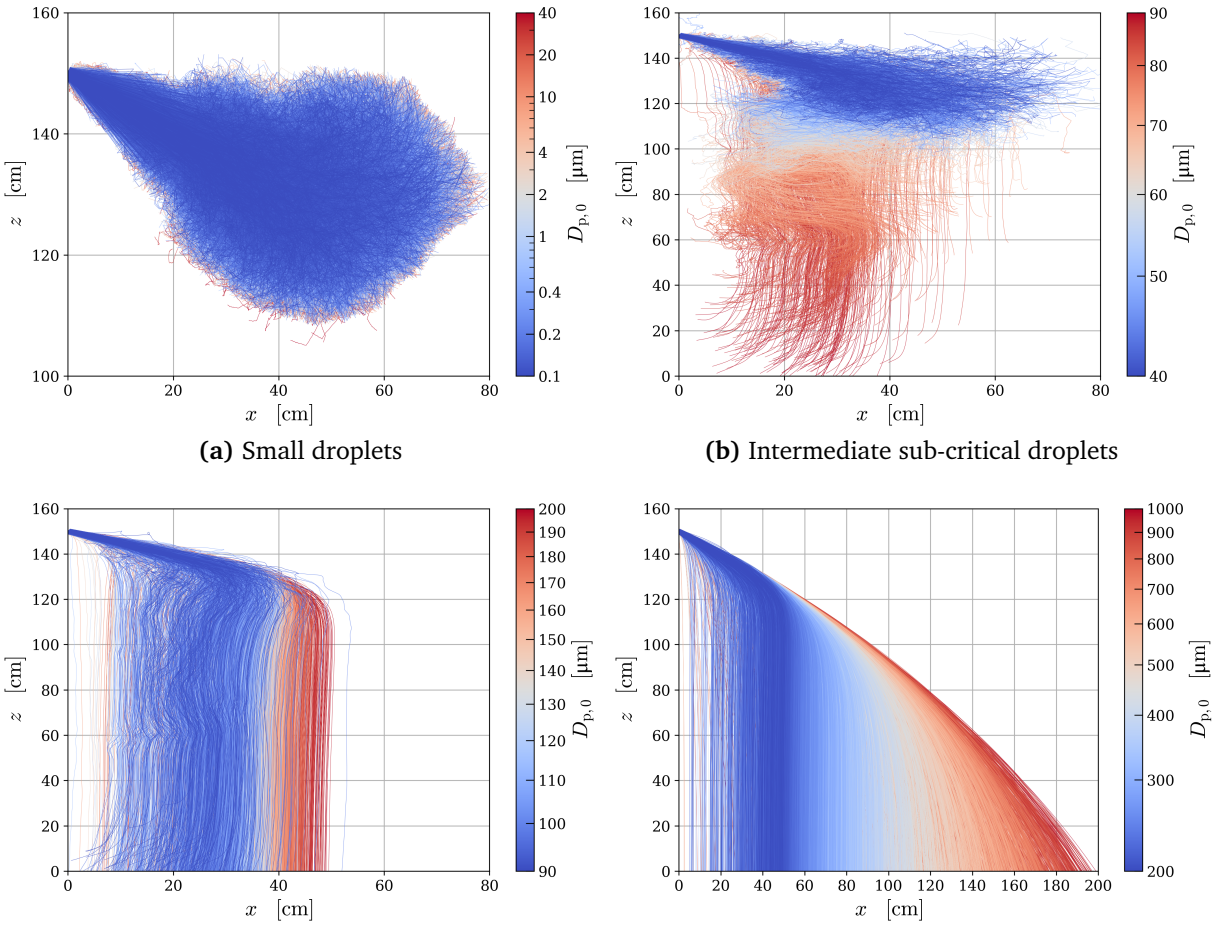
(b) absolute value of the y coordinate

Figure 5.10: Average position reached at the end of the simulation by droplet diameter; the simulation ends when the droplet is completely evaporated or touches the ground.

the longer time required for these droplets to complete evaporation and become airborne. It is worth reminding that the tracking of suspended droplets continues until they complete the evaporation. During the evaporation process of sub-critical intermediate droplets, the trajectory is increasingly dominated by drag; consequently, they are more affected by local air streams. On the other hand, super-critical droplets show an opposite trend: their dispersion in the y direction decreases with size, as larger droplets require less time to reach the ground. The minimum final y coordinate is observed for droplets of 300 µm, after which the value remains constant, as transport is negligible in this range. The black line in Figure 5.10b results from the extrapolation of the final trend, marking 200 µm as the threshold beyond which the transport by the air is negligible.

Finally, Figure 5.11 displays the same trajectories presented in Figure 5.9a, but separated into the distinct diameter ranges identified from the analysis of Figure 5.10. Figure 5.11a shows the trajectories of small droplets (up to 40 µm), which remain trapped within the breath cloud as they are fully transported by the airflow, with gravity having a negligible effect. In the image, it is also possible to observe some intermediate droplets leaving the cloud as they are also affected by gravity. Figure 5.11b shows the trajectories for sub-critical droplets, where it is possible to notice diverse behaviours: droplets with a diameter closer to 40 µm evaporate and become fully transported airborne particles, while droplets closer to 90 µm fall towards the ground following a winding path, affected by both gravity and transport. Super-critical droplets (Figure 5.11c) fall to the ground following a straighter path, as gravity starts to dominate over drag. Finally, Figure 5.11d reports the trajectories of droplets whose size is such that transport is negligible; in fact, they fall following an almost parabolic trajectory typical of particles falling by gravity.

To summarise, the quantitative analysis leads to the identification of 40 µm as the initial diameter below which gravity effects are negligible and transport dominates, and 200 µm



(a) Small droplets

(b) Intermediate sub-critical droplets

(c) Intermediate super-critical droplets

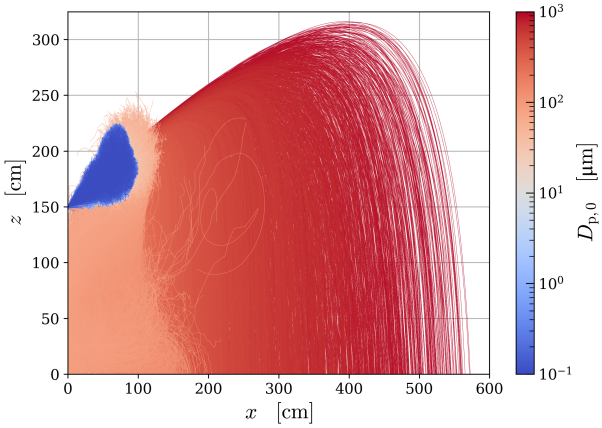
(d) Large droplets

Figure 5.11: Exhaled trajectories subdivided by droplet ranges.

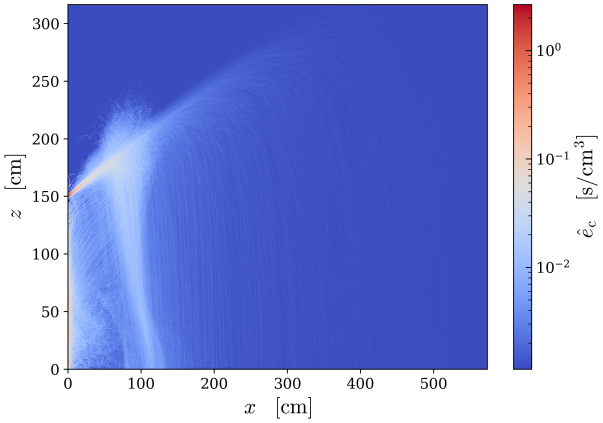
as the initial diameter above which transport is negligible and inertia dominates. These thresholds are not strict, since, as shown in Figure 5.11, the transition between different behaviours is smooth across the full range of diameters.

5.7.5. Sneezing

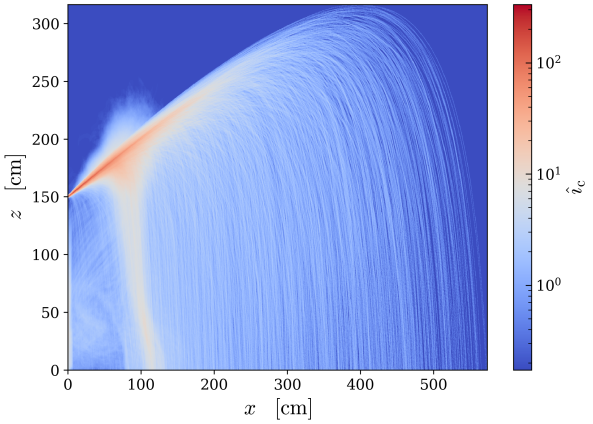
Figure 5.12 reports the results for the sneezing scenario. From Figure 5.12a, it is possible to make the same distinction between small, intermediate, and large droplets as done for the coughing scenario. Although the exhalation velocity is three times higher than in the coughing scenario, the breath cloud penetration is only slightly deeper, while the horizontal distance reached by larger droplets grows significantly, reaching 6 m. This is also due to the inclination of the exhalation, which is downward for the coughing scenario but upward for sneezing. From Figures 5.12b and 5.12c, it can be noted how falling intermediate-sized droplets create a region of high concentration at $\approx$ 1 m from the individual. However, a high risk of infection (Figure 5.12d) occurs also at greater distances due to the exhalation characteristics (high velocity tilted upward) and the large number of emitted *quanta*. As



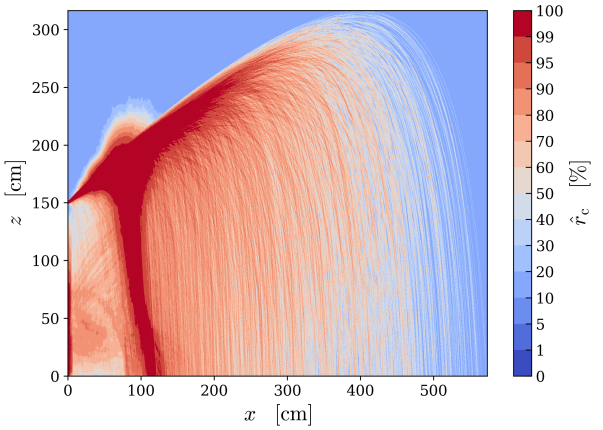
(a) Exhaled droplet trajectories coloured by droplet initial size



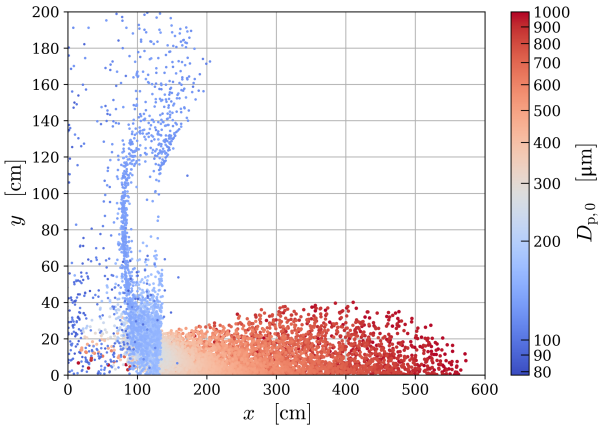
(b) Viral exposure map



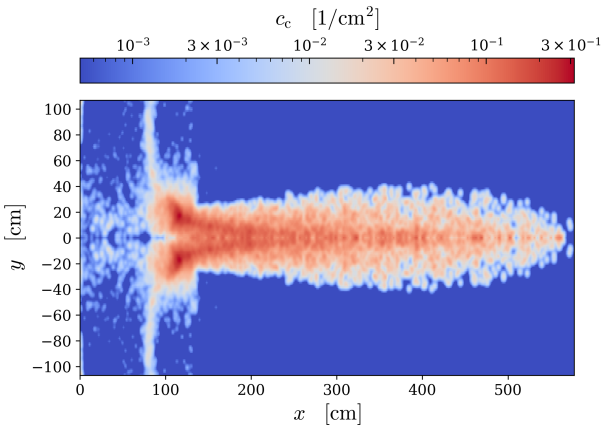
(c) Viral inhalation map after one hour exposure



(d) Risk of infection map after one hour exposure



(e) Exhaled droplet ground deposition coloured by droplet initial size



(f) Viral concentration at the ground after one hour exposure

Figure 5.12: Results of the sneezing scenario.

previously discussed, intermediate-sized droplets are transported by the airflow during evaporation before reaching the ground or becoming airborne. In Figure 5.12e, there is a high dispersion in the y direction for intermediate droplets; this is attributed to a large elongated air vortex occurring in front of the mannequin $\approx 4\,\mathrm{s}$ after the sneeze. This behaviour is most likely a spurious numerical effect of the CFD simulation, where a mostly quiescent environment is perturbed by high acceleration. However, even if it is only a numerical artifact, this demonstrates how a ventilation stream can significantly affect the trajectories of exhaled droplets. Apart from this, the ground deposition distribution (Figure 5.12f) is very similar to that of the coughing scenario, but with a larger surface area. In this case, the critical diameter resulting from the simulation is $108\,\mu\mathrm{m}$, as reported in Table 5.5.

5.8. Concluding remarks

A one-way coupled multi-scale model for the simulation of respiratory droplet transport and evaporation has been presented. The analysis relies on an analytical model for solving the governing equations of droplets motion and evaporation and on a series of U-RANS CFD simulations of the humid air stream exhaled during different types of respiratory events. This approach provides the analytical model with the required boundary conditions about the local environment encountered by the droplet during its flight.

Purely analytical models struggle to obtain reliable methods for assessing local airflow conditions, while purely numerical models, such as Eulerian-Lagrangian CFD simulations, are computationally expensive and limited in the number of discrete particles they can handle. The proposed multi-scale model aims to combine the efficient and accurate prediction of droplet transport and evaporation provided by the analytical tool with the ability of CFD to model inherently 3D and unsteady flow features. The analytical tool implements a discrete random walk model to account for turbulent droplet dispersion, as well as a randomisation of the droplet release within the mouth/nose cross-section and the respiratory event time span. These elements allow statistical analyses to be performed over the trajectories of large sets of droplets. The model was used to simulate large sets of over 40 000 droplets in five scenarios: mouth breathing, nose breathing, speaking, coughing, and sneezing. The results highlighted three main droplet behaviours depending on size. Small droplets evaporate rapidly, becoming airborne; their viral content is low and quickly dispersed in the environment. Intermediate-sized droplets are characterised by higher viral loads; they remain suspended for longer periods and are easily carried by local air currents, potentially creating localised areas of high viral concentration. Large droplets tend to follow parabolic trajectories before settling on a surface; they constitute the core of fomite transmission risk. By post-processing the trajectory data with assumptions on viral load and infectivity, quantitative risk maps were derived for direct inhalation, airborne transmission, and surface contamination. These maps provide a useful tool for quantifying the risk of contagion across different transmission routes.

Finally, it is important to emphasize that the analysis presented in this chapter was conducted in a generic, indoor environment. This test case serves as a preliminary validation of the multi-scale methodology. The results obtained here provide a fundamental understanding of the physical phenomena and droplet behaviours. The next step is to apply the model to the complex and confined geometry of a vehicle cabin. This analysis will be useful to investigate the influence of the HVAC system operating conditions on droplet dispersion and to assess the power consumption required to ensure the necessary air renewal rate. Moreover, in such an environment, higher droplet deposition is expected; consequently, the fomite transmission route becomes more relevant compared to the test case investigated in the present chapter.

Chapter 6

Conclusions

The commercialisation of Battery Electric Vehicles (BEVs) is growing year by year. However, one of the main bottlenecks to increasing the energy density, and consequently the total capacity of the batteries, is the temperature reached by the battery cells during operation. Researchers are focusing on various Battery Thermal Management Systems (BTMSs) such as air cooling, liquid cooling, phase change materials, heat pipes, and hybrid methods. However, current commercial BEVs with a capacity exceeding 62 kWh rely exclusively on liquid cooling for the batteries, while vehicles with lower capacity implement air cooling. For the preliminary design of the thermal management system of a BEV, the use of numerical models is essential, as they offer a flexible framework to study the effect of different thermal management strategies across many scenarios. To correctly model the thermal behaviour of a BEV, it is important to consider the relationship between phenomena occurring at different scales, from the electrochemical reaction scale of individual cells to the heat transfer throughout the entire battery pack, or in the case of integrated systems, throughout the entire vehicle, including the cabin and the motor block. This thesis deals with multiscale modelling to assist the thermal design of Battery Electric Vehicles (BEVs) and proposes a novel reduced-order approach for the optimisation of cooling channels layout.

Chapter 1 introduces the working principles of Lithium-ion batteries, focusing on their dependence on temperature to operate at maximum efficiency and avoid rapid degradation. Indeed, battery cells have an optimal operating range of 15–35 °C. Lower temperatures decrease the kinetics of internal reactions and favour lithium plating, which reduces battery capacity and increases internal impedance. Higher temperatures favour the growth of the Solid Electrolyte Interphase (SEI), which also causes capacity losses and increases internal resistance. Furthermore, if the temperature exceeds the critical limit, thermal runaway can occur, posing a risk to passenger safety. Subsequently, the chapter reviews the most

common strategies for the total thermal management system of BEVs, focusing on the BTMS, which is the most critical component given the restricted window of optimal operating temperatures for batteries. The final part of the chapter discusses the trade-off between the energy consumption of the Heating Ventilation and Air Conditioning (HVAC) system and the air quality within the cabin. To conserve energy, the recirculation mode can be utilised; however, this increases the concentration of pollutants, CO_2, and pathogens inside the cabin. Conversely, maintaining a high air change rate increases air quality but decreases the vehicle range.

The development of the numerical tool used as the basis for the analysis carried out in this thesis is presented in Chapter 2. This numerical tool can be applied to various engineering problems, and its modular implementation facilitates interfacing with other models to develop multiscale frameworks. The tool is based on domain representation via nodes connected by branches, with their relations described using graph theory. Based on this framework, different solvers can be implemented by solving the governing equations of the physics studied within the domain. The source terms of the balance equations are linearised, and iterative procedures are employed if non-linearities are present. To prevent common errors arising from unit mismatches in user input parameters, a unit of measurement management system is included in the tool, and, before solving the equations, the solvers check for unit consistency. This numerical tool currently integrates a pipe flow solver and a thermal network solver.

The thermal network solver is employed in Chapter 3 to develop a multiscale model to assess the operating temperatures of two battery packs integrated with a DC-DC power converter. This integrated system is cooled by means of a cooling plate and is intended for a high-power passenger BEV. The multiscale model discretises the two battery packs at the module level, where each pack comprises 248 modules, while the control boards are discretised at the switches level. The model couples the full-scale thermal network describing the overall heat transfer of the system, with an electrochemical model working at the electrochemical reaction scale. The convective heat transfer of the plate was extracted from a Computational Fluid Dynamics (CFD) simulation of a small periodic portion of the cooling plate, under the assumption of fully developed flow. This avoids the need to model the entire volume of the cooling plate, thereby reducing the computational burden. The case study analysed is based on the Worldwide Harmonised Light Vehicle Test Procedure (WLTP), which mimics vehicle behaviour under different driving conditions. The results lead to the conclusion that all components operate within safe conditions. Specifically, the battery packs present a mean temperature of $\sim 23\,^{\circ}\text{C}$ during the simulated time, with two spikes corresponding to current peaks, reaching $\sim 35\,^{\circ}\text{C}$.

The cooling plate used in the simulation of the integrated system contains standard serpentine channels derived from an initial draft design. To optimise the cooling plate per-

formance, a reduced-order optimisation method is proposed in Chapter 4. The optimisation method is based on Ant Colony Optimisation (ACO), a probabilistic meta-heuristic algorithm inspired by the behaviour of ants searching for the best path to food sources. In this specific case, the goal is to find the optimal path for the cooling channels to minimise the average temperature of the plate's top surface and increase temperature uniformity. To the best of the author's knowledge, this is the first application of ACO for such a problem. Due to the specific challenges of channels layout optimisation compared to standard ACO applications, the proposed algorithm extends the original method. It incorporates additional elements to improve the balance between exploration and exploitation, alongside constraints on the maximum pressure drop, maximum temperature standard deviation, and the number of fluid nodes. The algorithm starts from an empty grid of nodes and searches for the optimal paths connecting the inlet to the outlet. During the optimisation procedure, the generated solutions are evaluated using the thermal network solver detailed in Section 2.7. Subsequently, both the optimised solution and the initial configuration are simulated using 3D CFD models to obtain a more accurate estimation of the objective function improvement. The optimised solution decreases the average temperature of the plate's top surface by 27% and lowers the standard deviation by 7%. The head losses increase by a factor of 2.72 due to the higher number of flow obstacles and the longer channel length resulting from the optimisation. However, the pressure drop remains within the specified constraint. The strengths of this approach lie in its robustness and straightforward implementation compared to higher-order methods based on CFD or Finite Element Analysis. The proposed approach is not intended to substitute high-order topology optimisation. Instead, it serves as a computationally inexpensive method to generate simpler optimised solutions that are easily manufacturable or that can be used as initial configurations for higher-order optimisations.

Chapter 5 focuses on air quality and the infection risk due to respiratory droplets. The airborne and fomite infection risk in indoor environments under different respiratory events (mouth breathing, nose breathing, speaking, coughing, sneezing) is assessed by means of statistical analyses based on the results of a multiscale model. The multiscale model uses an analytical model to calculate droplet trajectories (with diameters between 1 mm and 0.1 μm) with CFD simulations of the exhaled airflow. Since the analytical model is limited in predicting airflow macro-features, 3D CFD simulations are used to provide the local ambient conditions required for the analytical model to solve the particle balance equations. The capabilities of the model are presented in a base case study involving a SARS-CoV-2 infected individual in an indoor environment. The results highlight three main droplet behaviours: larger droplets ($\gtrsim 200\,\mu m$) fall to the ground as they are mostly influenced by gravity due to their inertia; conversely, smaller droplets ($\lesssim 40\,\mu m$) have negligible inertia and their trajectories are influenced mostly by drag effects, with the possibility of being carried by the local airflow for long distances; between large and small droplets, there are

intermediate-sized droplets which possess characteristics of both behaviours, and depending on their specific history they can either fall to the ground or evaporate to become airborne particles and remain suspended like the small droplets. Among the simulated scenarios, sneezing is the worst case, as it is an event characterised by violent droplet release capable of throwing droplets up to a distance of 6m. Within 1 m distance from the infected individual, the infection risk is in the range of 90%–100%. The second worst-case scenario is coughing, which also exhibits high infection risk near the individual, roughly between 60%–100%. It is worth specifying that these values represent a single isolated event; with multiple events, the droplet concentration would increase, further increasing the infection risk. Expanding these results to the case of a vehicle cabin, highlights the importance of the correct design of the HVAC system to guarantee proper air quality. Vehicle cabins are much smaller than the volume considered in the test case and more passengers may be present; therefore, with a coughing or sneezing infected individual in the vehicle cabin, the infection risk would approach 100% without proper ventilation.

Future perspectives for the topics discussed in this thesis include the expansion of the numerical tool presented in Chapter 2 through the integration of a multi-physics solver, a dynamic time-stepping procedure, and the compilation of specific solver components to enhance computational efficiency. The multiscale methodology employed in Chapter 3 will be applied to perform additional analyses evaluating different scenarios and cooling strategies. The reduced-order optimisation discussed in Chapter 4 will be further investigated by expanding the parameter sensitivity analysis and testing new strategies to improve the balance between exploration and exploitation. Future efforts will also focus on refining the heuristic function and incorporating the optimisation of the channels aspect ratio and the number of parallel channels. Finally, the multiscale model utilised in Chapter 5 will be employed to analyse the infection risk within a vehicle cabin under various HVAC ventilation strategies, providing insights into the associated energy consumption.

Acknowledgements

PhD thesis funded by the European Union - NextGenerationEU under the National Recovery and Resilience Plan (PNRR) - Mission 4 “Education and research” - Component 2 “From research to business” - Investment 1.4 “Strengthening of research facilities and creation of “national R&D champions” on certain Key Enabling Technologies”.

The study presented in Chapter 3 received additional funding by the European Union’s Horizon Europe research and innovation program under grant agreement No. 101056781. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

APPENDIX A

Numerical methods

A.1. Upwind differencing

The Upwind Difference Scheme is a numerical method used to approximate the spatial derivatives in partial differential equations, particularly those containing advective terms, such as those in fluid dynamics and heat transfer. The core principle of the scheme is to use information from the direction of the flow (the "upwind" direction) when approximating the derivative at a given point. The best advantage of this scheme is its easy formulation and its high stability. In the solver developed in 2.6, the upwind scheme is used to transfer the fields of temperature, density, specific heat, and pressure, defined on the nodes, to the branches. The upwind scheme can be derived from a Taylor expansion: considering a generic grid of points distant Δx from each other and a generic field φ on the grid, the Taylor expansion of φ at the point $x + \Delta x$ is:

$$\varphi(x+\Delta x) = \varphi(x) + \frac{d}{dx}\varphi(x)\,\Delta x + \frac{1}{2}\frac{d^2}{dx^2}\varphi(x)\,\Delta x^2 + \dots \tag{A.1}$$

The upwind scheme sets the value of $\varphi(x+\Delta x)$ equal to $\varphi(x)$, so the truncation error can be evaluated as:

$$E = \varphi(x+\Delta x) - \varphi(x) = \frac{d}{dx}\varphi(x)\,\Delta x + \frac{1}{2}\frac{d^2}{dx^2}\varphi(x)\,\Delta x^2 + \dots = o(\Delta x)\,. \tag{A.2}$$

Since the truncation error is in the order of $o(\Delta x)$, the upwind differencing scheme is a first order accurate numerical method.

A.2. Midpoint rule

The Midpoint Rule is a numerical quadrature technique employed to estimate the definite integral of a function, operating on the principle of partitioning the total integration domain into smaller, non-overlapping subdivisions. The method proceeds by sampling the function's value at the central point within each of these subdivisions, using this single function value to approximate the entire integral contribution of that segment by multiplying it by the subdivision's width.

Consider a generic definite integral:

$$\int_a^b f(x)\,dx\,, \tag{A.3}$$

assuming that c is the midpoint of the integration interval: $c = a + b/2$. The Taylor expansion of $f(x)$ on c is:

$$f(x) = f(c) + \frac{d}{dx} f(c)(x-c) + \frac{1}{2}\frac{d^2}{dx^2} f(c)(x-c)^2 + \dots \tag{A.4}$$

Substituting the function inside the integral in Equation (A.3) with Equation (A.4), the result is

$$\int_a^b f(x)\,dx = f(c)(b-a) + \frac{1}{24}\frac{d^2}{dx^2} f(c)(b-a)^3 + \dots \tag{A.5}$$

In Equation (A.5) the terms with odd powers of $(x-c)$ cancel out because the integration interval is symmetric with respect to c, meaning only the contribution from the even-power terms remains. At this point is possible to evaluate the discretisation error as:

$$E = \frac{1}{b-a}\int_a^b f(x)\,dx - f(c) = \frac{1}{24}\frac{d^2}{dx^2} f(c)(b-a)^2\,, \tag{A.6}$$

demonstrating that the midpoint rule introduces a second order discretisation error. Using the midpoint rule means assuming that the integral average is equal to the midpoint value; in 2.6 this method is used to solve volume and surface integrals, considering the integral average equal to the value at the centre of the volume or surface.

A.3. Implicit Euler method

The Implicit Euler Method, sometimes referred to as the Backward Euler method, is a first order numerical procedure used for solving ordinary differential equations (ODEs) and is particularly common in transient problems. Unlike explicit methods, which calculate the state at the next time step (t^{n+1}) based solely on the known state at the current time step (t^n), the Implicit Euler Method defines the derivative using the unknown state at t^{n+1}. This characteristic means that when the underlying differential equation is non-linear, the method requires solving a non-linear algebraic equation or a system of equations at each time step. Its key advantage is its unconditional stability for stiff systems, allowing the use of larger time steps compared to explicit methods without encountering numerical instabilities. To derive the Euler method, it is possible to start from a classical Cauchy problem [272]:

$$\begin{cases} \dfrac{d\,\phi(t)}{dt} = f(\phi(t),\, t) \quad \text{for } t > t_0 \\ \phi\,(t_0) = \phi_0 \end{cases} \tag{A.7}$$

The goal is to solve Equation (A.7) between two time steps, t^n and $t^{n+1} = t^n + \Delta t$. At time t^{n+1} Equation (A.7) becomes:

$$\frac{d\phi\left(t^{n+1}\right)}{dt} = f\left(\phi\left(t^{n+1}\right),\, t^{n+1}\right) . \tag{A.8}$$

The time derivative in the Left Hand Side (LHS) of Equation (A.8) can be evaluated through a Taylor expansion truncated at the third term:

$$\frac{d\phi\left(t^{n+1}\right)}{dt} = \frac{\phi\left(t^{n+1}\right) - \phi\left(t^n\right)}{\Delta t} + o\left(\Delta t\right) . \tag{A.9}$$

Substituting the LHS term of Equation (A.8) with the Right Hand Side (RHS) term of Equation (A.9), and isolating $\phi\left(t^{n+1}\right)$, it is obtained:

$$\phi\left(t^{n+1}\right) = \phi\left(t^n\right) + f\left(\phi\left(t^{n+1}\right),\, t^{n+1}\right)\Delta t + o\left(\Delta t^2\right) . \tag{A.10}$$

From here, consider ϕ^{n+1} the approximation of $\phi(t^{n+1})$ by neglecting the term $o\left(\Delta t^2\right)$ in Equation (A.10). The same derivation can be applied to the value at the previous time step, leading to $\phi^n \approx \phi(t^n)$. By substituting the function values with their approximation, it is obtained the final formulation of the implicit Euler method:

$$\phi^{n+1} = \phi^n + f\left(\phi^{n+1},\, t^{n+1}\right)\Delta t . \tag{A.11}$$

The method is implicit since ϕ^{n+1} appears in both sides of Equation (A.11), therefore to use the implicit Euler usually there is the necessary to find an approximation for ϕ^{n+1} on the RHS, or use iterative methods.

The local truncation error at the step $n+1$ is:

$$E^{n+1} = \frac{\phi\left(t^{n+1}\right) - \phi\left(t^{n}\right)}{\Delta t} - f\left(\phi\left(t^{n+1}\right), t^{n+1}\right) . \tag{A.12}$$

Equation (A.12) can be re-arranged by considering the first term in the RHS and using Equation (A.8):

$$\begin{aligned} E^{n+1} &= \frac{d}{dt}\phi(t^{n+1}) - \frac{\Delta t}{2}\frac{d^2}{dt^2}\phi(t^{n+1}) - \frac{d}{dt}\phi(t^{n+1}) \\ &= -\frac{\Delta t}{2}\frac{d^2}{dt^2}\phi\left(t^{n+1}\right) . \end{aligned} \tag{A.13}$$

Therefore, the modulus of error is:

$$\left|E^{n+1}\right| \leqslant \frac{\Delta t}{2}\max\left(\frac{d^2}{dt^2}\phi(t)\right) \quad \text{for } t_0 \leqslant t \leqslant t_{\text{final}} . \tag{A.14}$$

Equation (A.14) demonstrates that the implicit Euler method is first order accurate.

A.3.1. Newton-Raphson applied to Implicit Euler

Since ϕ^{n+1} appears on both sides of Equation (A.11), iterative methods are required to ensure accuracy. The simplest approach is to start with an initial guess for ϕ^{n+1} on the RHS of Equation (A.11) and use the expression to calculate a new value on the LHS. This calculated value then serves as the guess for the subsequent iteration. This procedure is repeated until the difference between the guess and the calculated value falls below a specified tolerance.

Alternatively, the Implicit Euler step can be solved using the Newton-Raphson method to find the root of the residual function $g\left(\phi^{n+1}\right)$, which is defined by rearranging Equation (A.11):

$$g\left(\phi^{n+1}\right) = \phi^{n+1} - \phi^{n} - f\left(\phi^{n+1}\right)\Delta t . \tag{A.15}$$

Compared to the simple approach, the Newton-Raphson method guarantees faster convergence and the ability to handle stiff problems. Figure A.1 illustrates the principle behind the Newton-Raphson method. To find the value of ϕ^{n+1} that satisfies $g\left(\phi^{n+1}\right)=0$, the algorithm starts with an initial guess ϕ_0^{n+1}. The subsequent value is found by identifying the point where the tangent line at ϕ_0^{n+1} intersects the horizontal axis, governed by the equation:

$$\phi_1^{n+1} = \phi_0^{n+1} - \frac{g\left(\phi_0^{n+1}\right)}{g'\left(\phi_0^{n+1}\right)} , \tag{A.16}$$

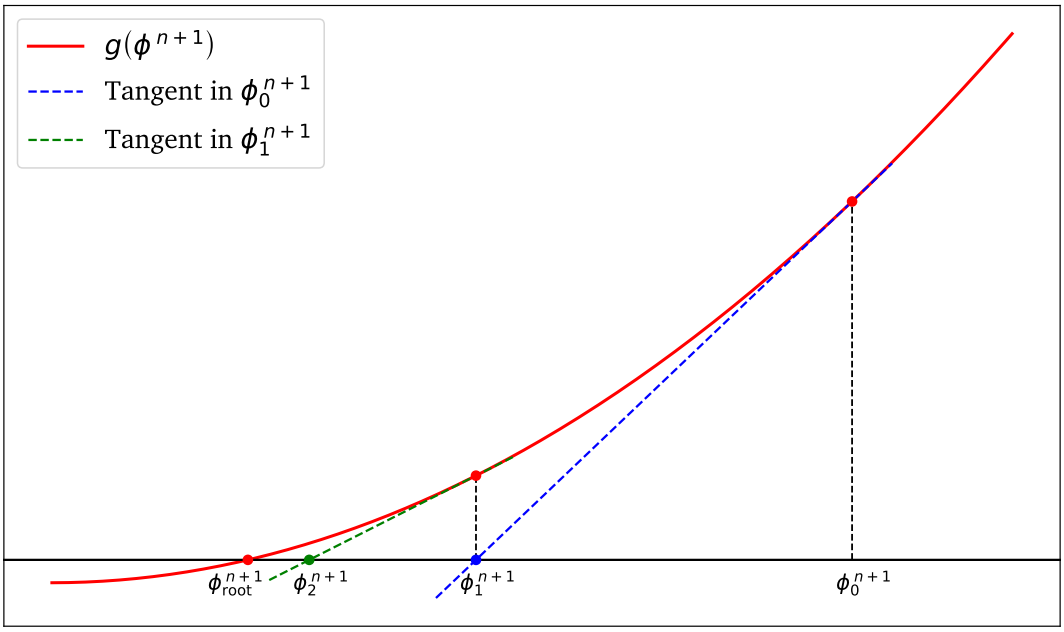


Figure A.1: Newton-Raphson algorithm; ϕ_0^{n+1} is the initial guess and ϕ_{root}^{n+1} is the actual root of $g\left(\phi^{n+1}\right)$.

where $g'\left(\phi_0^{n+1}\right)$ is the derivative of g evaluated at ϕ_0^{n+1} (*i.e.*, the slope of the tangent). The new value ϕ_1^{n+1} will be closer to the true root of the function, ϕ_{root}^{n+1}. If the derivative of g cannot be derived analytically, it can be approximated numerically by employing a small perturbation ε:

$$g'\left(\phi_0^{n+1}\right) = \frac{g\left(\phi_0^{n+1}+\varepsilon\right) - g\left(\phi_0^{n+1}\right)}{\varepsilon}\,. \tag{A.17}$$

This process is repeated iteratively to find new values of ϕ^{n+1} that progressively approach the true root. At the $(k+1)$-th iteration, the convergence error is defined as:

$$e_{k+1} = |\phi_{k+1}^{n+1} - \phi_k^{n+1}|\,. \tag{A.18}$$

The algorithm terminates when this error falls below a specified tolerance.

Bibliography

[1] IPCC. *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021.

[2] UNFCCC. The paris agreement. United Nations Framework Convention on Climate Change, 2015. Entry into force: 4 November 2016.

[3] European Union. Regulation (eu) 2021/1119 of the european parliament and of the council establishing the framework for achieving climate neutrality ('european climate law'). *Official Journal of the European Union*, L 243:1–17, 2021.

[4] European Environment Agency. Transport and environment report 2021: Decarbonising road transport — the role of vehicles, fuels and transport demand. Technical report, Publications Office of the European Union, Luxembourg, 2022.

[5] Eurostat. Greenhouse gas emissions from transport in the eu, 2023. Data retrieved from Eurostat Database.

[6] European Commission, Directorate-General for Mobility and Transport. *EU transport in figures: statistical pocketbook 2025*. Publications Office of the European Union, Luxembourg, 2025.

[7] European Commission. Fit for 55: delivering the eu's 2030 climate target on the way to climate neutrality. *COM/2021/550 final*, 2021.

[8] European Union. Regulation (eu) 2023/851 of the european parliament and of the council amending regulation (eu) 2019/631 as regards strengthening the co2

emission performance standards for new passenger cars and new light commercial vehicles. *Official Journal of the European Union*, L 110:5–20, 2023.

[9] S. F. Tie and C. W. Tan. A review of energy sources and energy management system in electric vehicles. *Renewable and Sustainable Energy Reviews*, 20:82–102, 2013.

[10] M. Ehsani, Y. Gao, S. Longo, and K. Ebrahimi. *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*. CRC Press, 3rd edition, 2018.

[11] C. C. Chan. The state of the art of electric, hybrid, and fuel cell vehicles. *Proceedings of the IEEE*, 95(4):704–718, 2007.

[12] S. Amjad, S. Neelakrishnan, and R. Rudramoorthy. Review of design considerations and technological challenges for successful development and deployment of plug-in hybrid electric vehicles. *Renewable and Sustainable Energy Reviews*, 14(3):1104–1110, 2010.

[13] M.A. Hannan, F.A. Azidin, and A. Mohamed. Hybrid electric vehicles and their challenges: A review. *Renewable and Sustainable Energy Reviews*, 29:135–150, 2014.

[14] B. G. Pollet, I. Staffell, and J. L. Shang. Current status of hybrid, battery and fuel cell electric vehicles: From electrochemistry to market prospects. *Electrochimica Acta*, 84:235–249, 2012.

[15] D.D. Friel. Chapter nineteen - management of batteries for electric traction vehicles. In Gianfranco Pistoia, editor, *Electric and Hybrid Vehicles*, pages 493–515. Elsevier, Amsterdam, 2010.

[16] Z. P. Cano, D. Banham, S. Ye, A. Hintennach, J. Lu, M. Fowler, and Z. Chen. Batteries and fuel cells for emerging electric vehicle markets. *Nature Energy*, 3(4):279–289, 2018.

[17] International Energy Agency. Global ev outlook 2023. Technical report, IEA, Paris, 2023.

[18] A. Masias, J. Marcicki, W. A. Paxton, and M. M. Mench. Thermal management of a li-ion battery pack at varying discharge rates. *Journal of Power Sources*, 259:206–211, 2013.

[19] A. A. Pesaran. Battery thermal models for hybrid vehicle simulations. *Journal of Power Sources*, 110(2):377–382, 2002.

[20] T. Waldmann, M. Wilka, M. Kasper, M. Fleischhammer, and M. Wohlfahrt-Mehrens. Temperature dependent ageing mechanisms in lithium-ion batteries – a post-mortem study. *Journal of Power Sources*, 262:129–135, 2014.

[21] Q. Wang, B. Jiang, B. Li, and Y. Yan. A critical review of thermal management models and solutions of lithium-ion batteries for the development of pure electric vehicles. *Renewable and Sustainable Energy Reviews*, 64:106–128, 2016.

[22] M. Ntombela, K. Musasa, and K. Moloi. A comprehensive review for battery electric vehicles (bev) drive circuits technology, operations, and challenges. *World Electric Vehicle Journal*, 14(7), 2023.

[23] S. Robert, A. Muhammed, and S. Günther. Comparatively assessing different shapes of lithium-ion battery cells. *Procedia Manufacturing*, 8:104–111, 2017. 14th Global Conference on Sustainable Manufacturing, GCSM 3-5 October 2016, Stellenbosch, South Africa.

[24] CRBAman. Lithium Ion Battery Cell - Cylindrical Cell, Prismatic Cell, Pouch Cell. `https://commons.wikimedia.org/wiki/File:Lithium_Ion_Battery_Cell_-_Cylindrical_Cell,_Prismatic_Cell,_Pouch_Cell.png`, July 2024. Wikimedia Commons. Licensed under CC BY-SA 4.0. Accessed: 2026-02-21.

[25] D.H. Jeon, S. Kim, and R. Hempelmann. State-of-the-art review of degradation mechanisms of commercial lithium-ion batteries. *Journal of Power Sources*, 646:237242, 2025.

[26] C. Lin, A. Tang, H. Mu, W. Wang, and C. Wang. Aging mechanisms of electrode materials in lithium-ion batteries for electric vehicles. *Journal of Chemistry*, 2015(1):104673, 2015.

[27] J. Vetter, P. Novák, M.R. Wagner, C. Veit, K.C. Möller, J.O. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, and A. Hammouche. Ageing mechanisms in lithium-ion batteries. *Journal of Power Sources*, 147(1):269–281, 2005.

[28] S. Bruno and G. Jürgen. Lithium batteries: Status, prospects and future. *Journal of Power Sources*, 195(9):2419–2430, 2010.

[29] M. Safari, M. Morcrette, A. Teyssot, and C. Delacourt. Multimodal physics-based aging model for life prediction of li-ion batteries. 156(3):A145, dec 2008.

[30] D. Witt, F. Röder, and U. Krewer. Analysis of lithium-ion battery state and degradation via physicochemical cell and sei modeling. *Batteries & Supercaps*, 5(7):e202200067, 2022.

[31] A. Tomaszewska, Z. Chu, X. Feng, S. O'Kane, X. Liu, J. Chen, C. Ji, E. Endler, R. Li, L. Liu, Y. Li, S. Zheng, S. Vetterlein, M. Gao, J. Du, M. Parkes, M. Ouyang, M. Marinescu, G. Offer, and B. Wu. Lithium-ion battery fast charging: A review. *eTransportation*, 1:100011, 2019.

[32] C. Fang, T.N. Tran, Y. Zhao, and G. Liu. Electrolyte decomposition and solid electrolyte interphase revealed by mass spectrometry. *Electrochimica Acta*, 399:139362, 2021.

[33] T. Kim, L.K. Ono, and Y. Qi. Understanding the active formation of a cathode–electrolyte interphase (cei) layer with energy level band bending for lithium-ion batteries. *J. Mater. Chem. A*, 11:221–231, 2023.

[34] P. Niehoff and P. Winter. Composition and growth behavior of the surface and electrolyte decomposition layer of/on a commercial lithium ion battery lixni1/3mn1/3co1/3o2 cathode determined by sputter depth profile x-ray photoelectron spectroscopy. *Langmuir : the ACS journal of surfaces and colloids*, 29:15813–21, 2013.

[35] S. Hein and A. Latz. Influence of local lithium metal deposition in 3d microstructures on local and global behavior of lithium-ion batteries. *Electrochimica Acta*, 201:354–365, 2016.

[36] C. Hendricks, N. Williard, C. Mathew, and M. Pecht. A failure modes, mechanisms, and effects analysis (fmmea) of lithium-ion batteries. *Journal of Power Sources*, 297:113–120, 2015.

[37] A. Mukhopadhyay and B.W. Sheldon. Deformation and stress in electrode materials for li-ion batteries. *Progress in Materials Science*, 63:58–116, 2014.

[38] X.H. Liu, L. Zhong, S. Huang, S.X. Mao, T. Zhu, and J.Y. Huang. Size-dependent fracture of silicon nanoparticles during lithiation. *ACS Nano*, 6(2):1522–1531, 2012.

[39] Y. Dai and A. Panahi. Thermal runaway process in lithium-ion batteries: A review. *Next Energy*, 6:100186, 2025.

[40] X. Feng, M. Ouyang, X. Liu, L. Lu, Y. Xia, and X. He. Thermal runaway mechanism of lithium ion battery for electric vehicles: A review. *Energy Storage Materials*, 10:246–267, 2018.

[41] R. Spotnitz and J. Franklin. Abuse behavior of high-power, lithium-ion cells. *Journal of Power Sources*, 113(1):81–100, 2003.

[42] Q. Wang, P. Ping, X. Zhao, G. Chu, J. Sun, and C. Chen. Thermal runaway caused fire and explosion of lithium ion battery. *Journal of Power Sources*, 208:210–224, 2012.

[43] H. Wang, Z. Du, X. Rui, S. Wang, C. Jin, L. He, F. Zhang, Q. Wang, and X. Feng. A comparative analysis on thermal runaway behavior of li (nixcoymnz) o2 battery with different nickel contents at cell and module level. *Journal of Hazardous Materials*, 393:122361, 2020.

[44] Y. Huang, Y. Dong, S. Li, J. Lee, C. Wang, Z. Zhu, W. Xue, Y. Li, and J. Li. Lithium manganese spinel cathodes for lithium-ion batteries. *Advanced Energy Materials*, 11(2):2000997, 2021.

[45] W. Wu, S. Wang, W. Wu, K. Chen, S. Hong, and Y. Lai. A critical review of battery thermal performance and liquid based battery thermal management. *Energy Conversion and Management*, 182:262–281, 2019.

[46] Z. Wang, J. Yuan, X. Zhu, H. Wang, L. Huang, Y. Wang, and S. Xu. Overcharge-to-thermal-runaway behavior and safety assessment of commercial lithium-ion cells with different cathode materials: A comparison study. *Journal of Energy Chemistry*, 55:484–498, 2021.

[47] X. Liu, D. Ren, H. Hsu, X. Feng, G.L. Xu, M. Zhuang, H. Gao, L. Lu, X. Han, Z. Chu, et al. Thermal runaway of lithium-ion batteries without internal short circuit. *Joule*, 2(10):2047–2064, 2018.

[48] F. Herrmann and F. Rothfuss. 1 - introduction to hybrid electric vehicles, battery electric vehicles, and off-road electric vehicles. In Bruno Scrosati, Jürgen Garche, and Werner Tillmetz, editors, *Advances in Battery Technologies for Electric Vehicles*, Woodhead Publishing Series in Energy, pages 3–16. Woodhead Publishing, 2015.

[49] T.J. Shelly, J.A. Weibel, D. Ziviani, and E.A. Groll. A dynamic simulation framework for the analysis of battery electric vehicle thermal management systems. In *2020 19th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, pages 538–546, 2020.

[50] J. Wang, S. Gao, J. Zhu, and J. Mao. Thermal performance analysis and burning questions of refrigerant direct cooling for electric vehicle battery. *Applied Thermal Engineering*, 232:121055, 2023.

[51] J. Guo and F. Jiang. A novel electric vehicle thermal management system based on cooling and heating of batteries by refrigerant. *Energy Conversion and Management*, 237:114145, 2021.

[52] Z. Zhu, Y. Zhang, A. Chen, J. Chen, Y. Wu, X. Wang, and T. Fei. Review of integrated thermal management system research for battery electrical vehicles. *Journal of Energy Storage*, 106:114662, 2025.

[53] W. Li, Y. Zhou, H. Zhang, and X. Tang. A review on battery thermal management for new energy vehicles. *Energies*, 16(13):4845, 2023.

[54] M. Henke and G. Hailu. Thermal management of stationary battery systems: A literature review. *Energies*, 13(16), 2020.

[55] Y. Fan, Y. Bao, C. Ling, Y. Chu, X. Tan, and S. Yang. Experimental study on the thermal management performance of air cooling for high energy density cylindrical lithium-ion batteries. *Applied Thermal Engineering*, 155:96—-109, 2019.

[56] X.M. Xu and R. He. Research on the heat dissipation performance of battery pack based on forced air cooling. *Journal of Power Sources*, 240:33–41, 2013.

[57] M. Wang, S. Teng, H. Xi, and Y. Li. Cooling performance optimization of air-cooled battery thermal management system. *Applied Thermal Engineering*, 195:117242, 2021.

[58] H. Sun and R. Dixon. Development of cooling strategy for an air cooled lithium-ion battery pack. *Journal of Power Sources*, 272:404–414, 2014.

[59] T. Yang, N. Yang, X. Zhang, and G. Li. Investigation of the thermal performance of axial-flow air cooling for the lithium-ion battery pack. *International Journal of Thermal Sciences*, 108:132–144, 2016.

[60] EV Specifications. Electric vehicles specifications database, 2026.

[61] J. Xu, Z. Guo, Z. Xu, X. Zhou, and X. Mei. A systematic review and comparison of liquid-based cooling system for lithium-ion batteries. *eTransportation*, 17:100242, 2023.

[62] D. Chen, J. Jiang, G.H. Kim, C. Yang, and A. Pesaran. Comparison of different cooling methods for lithium ion battery cells. *Applied Thermal Engineering*, 94:846–854, 2016.

[63] B. Tong, J. Shi, M. Cao, W. Xuan, J. Chen, K. Jin, J. Sun, and Q. Wang. Comprehensive comparison study on battery thermal management modules with indirect and direct liquid cooling. *Applied Thermal Engineering*, 268:125945, 2025.

[64] Anisha and A. Kumar. Identification and mitigation of shortcomings in direct and indirect liquid cooling-based battery thermal management system. *Energies*, 16(9), 2023.

[65] S. Panchal, R. Khasow, I. Dincer, M. Agelin-Chaab, R. Fraser, and M. Fowler. Thermal design and simulation of mini-channel cold plate for water cooled large sized prismatic lithium-ion battery. *Applied Thermal Engineering*, 122:80–90, 2017.

[66] A. Jarrett and I.Y. Kim. Design optimization of electric vehicle battery cooling plates for thermal performance. *Journal of Power Sources*, 196(23):10359–10368, 2011.

[67] Z. Wang, Z. Zou, Y. Zhou, X. Geng, Y. Sun, X. Huang, and M. Hao. Performance comparison of battery cold plates designed using topology optimization across laminar and turbulent flow regime. *International Journal of Heat and Mass Transfer*, 238:126450, 2025.

[68] H. Gao, X. Hou, W. Ma, and Y. Ma. Design and thermal performance analysis of a liquid cooling plate based on gradually varied circular notched fins for lithium-ion batteries. *Processes*, 13(3), 2025.

[69] T. Zhang, Q. Gao, G. Wang, Y. Gu, Y. Wang, W. Bao, and D. Zhang. Investigation on the promotion of temperature uniformity for the designed battery pack with liquid flow in cooling process. *Applied Thermal Engineering*, 116:655–662, 2017.

[70] M. Murphy and M. Akrami. Advanced thermal management of cylindrical lithium-ion battery packs in electric vehicles: A comparative cfd study of vertical, horizontal, and optimised liquid cooling designs. *Batteries*, 10(8), 2024.

[71] M. Falcone, E. Palka Bayard De Volo, A. Hellany, C. Rossi, and B. Pulvirenti. Lithium-ion battery thermal management systems: A survey and new cfd results. *Batteries*, 7(4), 2021.

[72] Y. Huo and Z. Rao. The numerical investigation of nanofluid based cylinder battery thermal management using lattice boltzmann method. *International Journal of Heat and Mass Transfer*, 91:374–384, 2015.

[73] X.H. Yang, S.C. Tan, and J. Liu. Thermal management of li-ion battery with liquid metal. *Energy Conversion and Management*, 117:577–585, 2016.

[74] M. Al-Zareer, I. Dincer, and M.A. Rosen. Electrochemical modeling and performance evaluation of a new ammonia-based battery thermal management system for electric and hybrid electric vehicles. *Electrochimica Acta*, 247:171–.182, 2017.

[75] A. Calborean, L. Máthé, and O. Bruj. Phase change materials for thermal management in lithium-ion battery packs: A review. *Batteries*, 11(12), 2025.

[76] W. Wu, X. Yang, G. Zhang, X. Ke, Z. Wang, W. Situ, X. Li, and J. Zhang. An experimental study of thermal management system using copper mesh-enhanced composite phase change materials for power battery pack. *Energy*, 113:909–916, 2016.

[77] T. Oya, T. Nomura, M. Tsubota, N. Okinaka, and T. Akiyama. Thermal conductivity enhancement of erythritol as pcm by using graphite and nickel particles. *Applied Thermal Engineering*, 61(2):825–828, 2013.

[78] Y. Lv, W. Situ, X. Yang, G. Zhang, and Z. Wang. A novel nanosilica-enhanced phase change material with anti-leakage and anti-volume-changes properties for battery thermal management. *Energy Conversion and Management*, 163:250–259, 2018.

[79]

[80] J. Liu, H. Chen, S. Huang, Y. Jiao, and M. Chen. Recent progress and prospects in liquid cooling thermal management system for lithium-ion batteries. *Batteries*, 9(8), 2023.

[81] M. Bernagozzi, A. Georgoulas, N. Miché, and M. Marengo. Heat pipes in battery thermal management systems for electric vehicles: A critical review. *Applied Thermal Engineering*, 219:119495, 2023.

[82] H. Akachi. Structure of a heat pipe, May 1990.

[83] L. Pagliarini, N. Iwata, and F. Bozzoli. Pulsating heat pipes: Critical review on different experimental techniques. *Experimental Thermal and Fluid Science*, 148:110980, 2023.

[84] A. Greco, D. Cao, X. Jiang, and H. Yang. A theoretical and computational study of lithium-ion battery thermal management for electric vehicles using heat pipes. *Journal of Power Sources*, 257:344–355, 2014.

[85] S. Deng, K. Li, Y. Xie, C. Wu, P. Wang, M. Yu, B. Li, and J. Zheng. Heat pipe thermal management based on high-rate discharge and pulse cycle tests for lithium-ion batteries. *Energies*, 12(16), 2019.

[86] L. Cattani, F. Sacchelli, and F. Bozzoli. Enhanced passive thermal management for electric vehicle batteries using a 3d pulsating heat pipe. *Energies*, 18(9), 2025.

[87] Z. Liang, R. Wang, A.H. Malt, M. Souri, M.N. Esfahani, and M. Jabbari. Systematic evaluation of a flat-heat-pipe-based thermal management: Cell-to-cell variations and battery ageing. *Applied Thermal Engineering*, 192:116934, 2021.

[88] M.A. Bamdezh and G.R. Molaeimanesh. The path from conventional battery thermal management systems to hybrid battery thermal management systems for electric vehicles, opportunities and challenges. *Journal of Energy Storage*, 100:113160, 2024.

[89] Y.S. Ranjbaran, M.H. Shojaeefard, and G.R. Molaeimanesh. Thermal performance enhancement of a passive battery thermal management system based on phase change material using cold air passageways for lithium batteries. *Journal of Energy Storage*, 68:107744, 2023.

[90] P. Qin, M. Liao, D. Zhang, Y. Liu, J. Sun, and Q. Wang. Experimental and numerical study on a novel hybrid battery thermal management system integrated forced-air convection and phase change material. *Energy Conversion and Management*, 195:1371–1381, 2019.

[91] Y. Lv, G. Liu, G. Zhang, and X. Yang. A novel thermal management structure using serpentine phase change material coupled with forced air convection for cylindrical battery modules. *Journal of Power Sources*, 468:228398, 2020.

[92] Z. An, X. Chen, L. Zhao, and Z. Gao. Numerical investigation on integrated thermal management for a lithium-ion battery module with a composite phase change material and liquid cooling. *Applied Thermal Engineering*, 163:114345, 2019.

[93] J. Xiao, H. Min, H. Jiang, Z. Zhang, W. Sun, and Q. Cao. A novel hybrid battery thermal management integrating phase change material and micro-channel liquid cooling. *Applied Thermal Engineering*, 274:126721, 2025.

[94] H. Liu, S. Ahmad, Y. Shi, and J. Zhao. A parametric study of a hybrid battery thermal management system that couples pcm/copper foam composite with helical liquid channel cooling. *Energy*, 231:120869, 2021.

[95] J. E, F. Yi, W. Li, B. Zhang, H. Zuo, K. Wei, J. Chen, H. Zhu, H. Zhu, and Y. Deng. Effect analysis on heat dissipation performance enhancement of a lithium-ion-battery pack with heat pipe for central and southern regions in china. *Energy*, 226:120336, 2021.

[96] C. Zhang, Z. Xia, B. Wang, H. Gao, S. Chen, S. Zong, and K. Luo. A li-ion battery thermal management system combining a heat pipe and thermoelectric cooler. *Energies*, 13(4), 2020.

[97] J. Smith, R. Singh, M. Hinterberger, and M. Mochizuki. Battery thermal management system for electric vehicle using heat pipes. *International Journal of Thermal Sciences*, 134:517–529, 2018.

[98] X. Yuan, A. Tang, C. Shan, Z. Liu, and J. Li. Experimental investigation on thermal performance of a battery liquid cooling structure coupled with heat pipe. *Journal of Energy Storage*, 32:101984, 2020.

[99] L. Cattani, M. Malavasi, F. Bozzoli, and C. Sciancalepore. Two-phase cooling system for electric vehicles' battery. 2766(1):012011, may 2024.

[100] M. Lesage, D. Chalet, J. Migaud, and C. Krautner. Optimization of air quality and energy consumption in the cabin of electric vehicles using system simulation. *Journal of Environmental Management*, 358:120861, 2024.

[101] J. Glos, L. Otava, and P. Václavek. Non-linear model predictive control of cabin temperature and air quality in fully electric vehicles. *IEEE Transactions on Vehicular Technology*, 70(2):1216–1229, 2021.

[102] L. Russi, P. Guidorzi, B. Pulvirenti, D. Aguiari, G. Pau, and G. Semprini. Air quality and comfort characterisation within an electric vehicle cabin in heating and cooling operations. *Sensors*, 22(2), 2022.

[103] S. Zong, W. Wang, X. Yin, Y. Song, L. Huang, F. Cao, Z. Zhang, and B. Wang. Evaluation of energy-saving potential and cabin thermal comfort for automobile co2 heat pump. *Applied Thermal Engineering*, 228:120339, 2023.

[104] Z. Liu, Y. Xie, X. Hu, B. Shi, and X. Lin. A control strategy for cabin temperature of electric vehicle considering health ventilation for lowering virus infection. *International Journal of Thermal Sciences*, 172:107371, 2022.

[105] P. Zhang, M. Djeddou, X. Su, G. Gao, and W. Lu. An impinging jet ventilation approach for automotive cabins: promoting coughing droplet mitigation and thermal comfort enhancement. *Applied Thermal Engineering*, 281:128696, 2025.

[106] R.M. Mariita, J.H. Davis, M.M. Lottridge, R.V. Randive, H. Witting, and J. Yu. Towards a healthy car: Uvc leds in an automobile's hvac demonstrates effective disinfection of cabin air. *Atmosphere*, 13(11), 2022.

[107] Simulink Documentation. Simulation and model-based design, 2020.

[108] M. Lebrun, D. Vasiliu, and N. Vasiliu. Numerical simulation of the fluid control systems by amesim. *Studies in Informatics and Control*, 18:111–118, 06 2009.

[109] GT-SUITE. Integrated multi-physics systems simulation.

[110] Altair Documentation. Thermo-fluid system design, 2025.

[111] G. Van Rossum, F.L. Drake, et al. *Python reference manual*, volume 111. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[112] S.A. Klein et al. Trnsys, a transient system simulation program manual, 2018.

[113] B. A. Cullimore. Sinda/fluint: Recent and on-going expansions of the industry-standard thermal/fluid analyzer. In *International Conference On Environmental Systems*. SAE International, July 1997.

[114] Thermal Desktop Documentation. Ansys thermal desktop heat transfer & fluid flow modeling software, 2025.

[115] P. Fritzson, A. Pop, K. Abdelhak, A. Ashgar, B. Bachmann, W. Braun, D. Bouskela, R. Braun, L. Buffoni, F. Casella, R. Castro, R. Franke, D. Fritzson, M. Gebremedhin, A. Heuermann, B. Lie, A. Mengist, L. Mikelsons, K. Moudgalya, L. Ochel, A. Palanisamy, V. Ruge, W. Schamai, M. Sjölund, B. Thiele, J. Tinnerholm, and P. Östlund. The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development. *Modeling, Identification and Control*, 41(4):241–295, 2020.

[116] L.R. Petzold. Dassl. a differential/algebraic system solver. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 1982.

[117] L.R. Petzold. Description of dassl: a differential/algebraic system solver. Technical report, Sandia National Labs., Livermore, CA (USA), 1982.

[118] D. Brück, H. Elmqvist, S.E. Mattsson, and H. Olsson. Dymola for multi-engineering modeling and simulation. In *Proceedings of modelica*, volume 2002. Citeseer, 2002.

[119] M. Tiller. *Introduction to physical modeling with Modelica*. Springer Science & Business Media, 2001.

[120] S.L. Campbell, J.P. Chancelier, R. Nikoukhah, S.L. Campbell, and R. Nikoukhah. *Modeling and Simulation in SCILAB*. Springer, 2010.

[121] A. C. Hindmarsh and L. R. Petzold. Lsoda, ordinary differential equation solver for stiff or non-stiff system, Sep 2005.

[122] S. D. Cohen, A. C. Hindmarsh, and P. F. Dubois. Cvode, a stiff/nonstiff ode solver in c. *Computer in Physics*, 10(2):138–143, 1996.

[123] A. C. Hindmarsh, R. Serban, C. J. Balos, D. J. Gardner, D. R. Reynolds, and C. S. Woodward. Implicit differential algebraic equations system solver, 2025.

[124] C.R. Harris et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[125] P. Virtanen et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[126] T. Brown, J. Hörsch, and D. Schlachtberger. Pypsa: Python for power system analysis. *arXiv preprint arXiv:1707.09913*, 2017.

[127] F. Witte and I. Tuschy. TESPy: Thermal Engineering Systems in Python. *Journal of Open Source Software*, 5(49):2178, 2020.

[128] I.H. Bell, J. Wronski, S. Quoilin, and V. Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, 53(6):2498–2508, 2014.

[129] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 54–73. Elsevier, 1983.

[130] G. Van Rossum and F.L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[131] Mishap Investigation Board. Mars climate orbiter mishap investigation board phase i report november 10, 1999.

[132] M. Grespan. *Thermal management of high power density components for sustainable mobility: development of computational methods and case studies*. PhD thesis, Università degli Studi di Modena e Reggio Emilia, Reggio Emilia, Italy, 2025.

[133] G. Cantore. *Macchine: appunti tratti dalle lezioni del Prof. Cantore*. Esculapio, 1999.

[134] C.F. Colebrook. Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws. *Journal of the Institution of Civil Engineers*, 11(4):133–156, 1939.

[135] S.E. Haaland. Simple and explicit formulas for the friction factor in turbulent pipe flow. *Journal of Fluids Engineering*, 105(1):89–90, 1983.

[136] J.D. Anderson. *Computational Fluid Dynamics*. Computational Fluid Dynamics: The Basics with Applications. McGraw-Hill Education, 1995.

[137] D. Dan, Y. Chengning, Z. Yangjun, Z. Hu, Z. Zezhi, and X. Xiaoming. Dynamic thermal behavior of micro heat pipe array-air cooling battery thermal management system based on thermal network model. *Applied Thermal Engineering*, 162:114183, 2019.

[138] M. Grespan, S. Busquets-Monge, E. Mas de les Valls, S. Alepuz, M. Raya, X. Jordà, D. Barater, and D. Angeli. Multiscale thermal modelling of integrated motor drives with pcb-embedded sic power devices. *Applied Thermal Engineering*, page 131094, 2026.

[139] D. Anshuman, S. Navid, L. Ri, E. Wilson, and K. Rahul. Boundary condition independent thermal network modeling of high-frequency power transformers. *Heat Transfer Engineering*, 44(3):259–276, 2023.

[140] Li Zhang, H. Zhao, S. Liang, and C. Liu. Heat transfer in phase change materials for integrated batteries and power electronics systems. *Applied Thermal Engineering*, 232:120997, 2023.

[141] A. Balachandran. *Battery Integrated Modular Multilevel Converter Topologies for Automotive Applications*. PhD thesis, 02 2023.

[142] R.M. Reddy and M. Das. A reconfigurable bidirectional dc–dc converter with integrated battery heating for electric vehicle applications. *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, 4(4):1181–1191, 2023.

[143] A. Balachandran, T. Jonsson, and L. Eriksson. Design and analysis of battery-integrated modular multilevel converters for automotive powertrain applications. In *2021 23rd European Conference on Power Electronics and Applications (EPE'21 ECCE Europe)*, pages 1–12, 2021.

[144] L. Zhang, H. Zhao, and C. Liu. Thermal management for transient integrated battery and power electronics systems using phase change materials. *International Journal of Thermal Sciences*, 209:109526, 2025.

[145] O. Alavi, T. Rajabloo, W. De Ceuninck, and M. Daenen. Non-isolated dc-dc converters in fuel cell applications: Thermal analysis and reliability comparison. *Applied Sciences*, 12(10), 2022.

[146] S. Wu, J. Zhou, X. Zhang, and J. Yu. Design and research on high power density motor of integrated motor drive system for electric vehicles. *Energies*, 15(10), 2022.

[147] X. Li, M. Yao, Q. Yang, M. Wang, and B. Yang. Modeling of an integrated drive unit in an electric vehicle. In *2020 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 125–127, 2020.

[148] A. Bourgault, O. Taqavi, Z. Li, G. Byczynski, and N.C. Kar. Advanced lumped parameter thermal network for modeling of cooling solutions in electric vehicle motor applications. *IEEE Transactions on Magnetics*, 60(11):1–5, 2024.

[149] P. Azer, R. Rodriguez, H. Ge, J. Bauman, P.S. Ravi, and A. Emadi. Time efficient integrated electro-thermal model for bidirectional synchronous dc-dc converter in hybrid electric vehicles. In *2018 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 55–62, 2018.

[150] Xulong Wang, Jiahao She, Bing Zhu, and Jun Yang. Thermal network modeling of insulated core transformers. In *2024 6th International Conference on Energy Systems and Electrical Power (ICESEP)*, pages 1109–1113, 2024.

[151] M. Alhanouti, M. Gießler, T. Blank, and F. Gauterin. New electro-thermal battery pack model of an electric vehicle. *Energies*, 9(7), 2016.

[152] N. Damay, C. Forgez, M.P. Bichat, and G. Friedrich. Thermal modeling of large prismatic lifepo4/graphite battery. coupled thermal and heat generation models for characterization and simulation. *Journal of Power Sources*, 283:37–45, 2015.

[153] V. Mannapperuma, L.C. Gaddala, R. Zheng, D. Kim, Y. Kim, A. Ullal, S. Zhu, and K.P. Ha. Electro-thermal modeling and parameter identification of an ev battery pack using drive cycle data. *Batteries*, 11(9), 2025.

[154] L.H. Saw, Y. Ye, and A.A.O. Tay. Electro-thermal analysis and integration issues of lithium ion battery for electric vehicles. *Applied Energy*, 131:97–107, 2014.

[155] C. Reiter, J. Dirnecker, and M. Lienkamp. Efficient simulation of thermal management systems for bev. In *2019 Fourteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, pages 1–8, 2019.

[156] K. Murashko, W. Huapeng, J. Pyrhönen, and L. Laurila. Modelling of the battery pack thermal management system for hybrid electric vehicles. In *2014 16th European Conference on Power Electronics and Applications*, pages 1–10, 2014.

[157] S. Bhattacharyya, Q.T. Dinh, and A. McGordon. Reduced-order thermal modelling of battery modules under passive air cooling. In *2025 IEEE/AIAA Transportation Electrification Conference and Electric Aircraft Technologies Symposium (ITEC+EATS)*, pages 1–6, 2025.

[158] R.A. Torres, H. Dai, T.M. Jahns, B. Sarlioglu, and W. Lee. Cooling design of integrated motor drives using analytical thermal model, finite element analysis, and computational fluid dynamics. In *2021 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 1509–1509, 2021.

[159] I. Aghabali, J. Bauman, and A. Emadi. Analysis of auxiliary power unit and charging for an 800v electric vehicle. In *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–6, 2019.

[160] J. Dong-Keun, H.S. Kim, J.W. Baek, J.Y. Kim, and H.J. Kim. Dual active bridge converter for energy storage system in dc microgrid. In *2016 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, pages 152–156, 2016.

[161] R.W.A.A. De Doncker, D.M. Divan, and M.H. Kheraluwala. A three-phase soft-switched high-power-density dc/dc converter for high-power applications. *IEEE Transactions on Industry Applications*, 27(1):63–73, 1991.

[162] N.S. Spinner, R. Mazurick, A. Brandon, S.L. Rose-Pehrsson, and S.G. Tuttle. 162(14):A2789, oct 2015.

[163] D. Staton, A. Boglietti, and A. Cavagnino. Solving the more difficult aspects of electric motor thermal analysis in small and medium size industrial induction motors. *IEEE Transactions on Energy Conversion*, 20(3):620–628, 2005.

[164] S. Busquets-Monge, A. Filba-Martinez, S. Alepuz, J. Nicolas-Apruzzese, A. Luque, A. Conesa-Roca, and J. Bordonau. Multibattery-fed neutral-point-clamped dc–ac converter with soc balancing control to maximize capacity utilization. *IEEE Transactions on Industrial Electronics*, 67(1):16–27, 2020.

[165] D. Bernardi, E. Pawlikowski, and J. Newman. A general energy balance for battery systems. 132(1):5, jan 1985.

[166] L. Jie, Y. Saurabh, S. Mohammad, C. Santosh, and S.C. Kim. Review of thermal coupled battery models and parameter identification for lithium-ion battery heat generation in ev battery thermal management system. *International Journal of Heat and Mass Transfer*, 218:124748, 2024.

[167] J. Guo, Q. Guo, J. Liu, and H. Wang. The polarization and heat generation characteristics of lithium-ion battery with electric–thermal coupled modeling. *Batteries*, 9(11), 2023.

[168] H. Shi, L. Wang, S. Wang, C. Fernandez, X. Xiong, B.E. Dablu, and W. Xu. A novel lumped thermal characteristic modeling strategy for the online adaptive temperature and parameter co-estimation of vehicle lithium-ion batteries. *Journal of Energy Storage*, 50:104309, 2022.

[169] N. Zatta, B. De Cesaro, E. Dal Cin, G. Carraro, G. Cristofoli, A. Trovò, A. Lazzaretto, and M. Guarnieri. Holistic testing and characterization of commercial 18650 lithium-ion cells. *Batteries*, 10(7), 2024.

[170] F.R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, 1994.

[171] M. Doninelli. *Le reti di distribuzione*. Caleffi S.p.A., 2002. Accessed 29/01/2026.

[172] Y. Huang, P. Mei, Y. Lu, R. Huang, X. Yu, Z. Chen, and A.P. Roskilly. A novel approach for lithium-ion battery thermal management with streamline shape mini channel cooling plates. *Applied Thermal Engineering*, 157:113623, 2019.

[173] M.P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.

[174] T. Borrvall and J. Petersson. Topology optimization of fluids in stokes flow. *International Journal for Numerical Methods in Fluids*, 41(1):77–107, 2003.

[175] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *International Journal for Numerical Methods in Fluids*, 58(8):861–877, 2008.

[176] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the spalart–allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, 2009.

[177] E.M. Dede. Multiphysics topology optimization of heat transfer and fluid flow systems. In *Proceedings of the COMSOL Conference*, Boston, MA, 2009.

[178] G.H. Yoon. Topological design of heat dissipating structure with forced convection. *Journal of Mechanical Science and Technology*, 24(6):1225–1233, 2010.

[179] E.M. Papoutsis-Kiachagias, E.A. Kontoleontos, A.S. Zymaris, D.I. Papadimitriou, and K.C. Giannakoglou. Constrained topology optimization for laminar and turbulent flows, including heat transfer. In C. Poloni, D. Quagliarella, J. Périaux, N. Gauger, and K. Giannakoglou, editors, *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems*, Capua, Italy, 2011. CIRA.

[180] E. A. Kontoleontos, E. M. Papoutsis-Kiachagias, A. S. Zymaris, D. I. Papadimitriou, and K. C. Giannakoglou. Adjoint-based constrained topology optimization for viscous flows, including heat transfer. *Engineering Optimization*, 45(8):941–961, 2013.

[181] M. Pietropaoli, F. Montomoli, and A. Gaymann. Three-dimensional fluid topology optimization for heat transfer. *Structural and Multidisciplinary Optimization*, 59(3):801–812, 2019.

[182] V. Subramaniam, T. Dbouk, and J.-L. Harion. Topology optimization of conjugate heat transfer systems: A competition between heat transfer enhancement and pressure drop reduction. *International Journal of Heat and Fluid Flow*, 75:165–184, 2019.

[183] F. Dugast, Y. Favennec, C. Josset, Y. Fan, and L. Luo. Topology optimization of thermal fluid flows with an adjoint lattice boltzmann method. *Journal of Computational Physics*, 365:376–404, 2018.

[184] Kentaro Yaji, Takayuki Yamada, Masato Yoshino, Toshiro Matsumoto, Kazuhiro Izui, and Shinji Nishiwaki. Topology optimization in thermal-fluid flow using the lattice boltzmann method. *Journal of Computational Physics*, 307:355–377, 2016.

[185] E. Gallorini, J. Hélie, and F. Piscaglia. A multi region adjoint-based solver for topology optimization in conjugate heat transfer problems. *Computers & Fluids*, 266:106042, 2023.

[186] E. Gallorini, J. Hélie, and F. Piscaglia. An adjoint-based solver with adaptive mesh refinement for efficient design of coupled thermal-fluid systems. *International Journal for Numerical Methods in Fluids*, 95(7):1065–1090, 2023.

[187] N. Galanos, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. The cut-cell method for the conjugate heat transfer topology optimization of turbulent flows using the "think discrete-do continuous" adjoint. *Energies*, 17(8):1817, 2024.

[188] M. Yu, S. Ruan, J. Gu, M. Ren, Z. Li, X. Wang, and C. Shen. Three-dimensional topology optimization of thermal-fluid-structural problems for cooling system design. *Structural and Multidisciplinary Optimization*, 62:3347–3366, 2020.

[189] W. Sun, P. Li, T. Zhou, Y. Li, C. Li, X. Shao, and H. Shen. Thermo-fluids performance analysis and experimental verification of topologically optimized mini-channel heat sinks integrated with impact jet. *International Journal of Thermal Sciences*, 211:109705, 2025.

[190] Y. Liu, C. Chen, Y. Yuan, J. Yang, Z. Guo, and J. Shi. Study of microchannel heat transfer characteristics based on topology optimization. *International Journal of Thermal Sciences*, 214:109898, 2025.

[191] P. Hou, F. Ma, J. Zhao, Q. Huang, H. Cui, J. Wang, and S. Wang. Experimental study on the heat transfer and flow characteristics of phase change slurry in topology optimization cold plate. *International Journal of Heat and Mass Transfer*, 256:127955, 2026.

[192] M.S. Wu. Multi-objective topology optimization of cold plates featuring branched and streamlined mini-channels for thermal management system of lithium-ion battery module. *Journal of Energy Storage*, 72:108362, 2023.

[193] T. Yang, H. Liu, W. Zhang, A. Ding, and M. Wu. Multi-objective topology optimization of the cooling plate for battery thermal management. *Batteries*, 11(11), 2025.

[194] Q. Zhong, A. Garg, L. Gao, A. Garg, B. Panda, and K. Wei. Topology-optimized liquid cooling plates for low-temperature battery preheating: A multi-objective thermal management strategy. *Thermal Science and Engineering Progress*, 67:104149, 2025.

[195] T. Dbouk. A review about the engineering design of optimal heat transfer systems using topology optimization. *Applied Thermal Engineering*, 112:841–854, 2017.

[196] A. Fawaz, Y. Hua, S. Le Corre, Y. Fan, and L. Luo. Topology optimization of heat exchangers: A review. *Energy*, 252:124053, 2022.

[197] J. Liu, R. Li, K. Wang, and J. Zheng. Net-based thermal-fluid model and hybrid optimization of cooling channels. *International Journal for Numerical Methods in Engineering*, (22):5442–5472, 2022.

[198] A. Pizzolato, A. Sciacovelli, and V. Verda. Topology optimization of robust district heating networks. *Journal of Energy Resources Technology*, 139(3), 2017.

[199] M. Blommaert, Y. Wack, and M. Baelmans. An adjoint optimization approach for the topological design of large-scale district heating networks based on nonlinear models. *Applied Energy*, 280:116025, 2020.

[200] P. Egberts, C. Tumer, K. Loh, and R. Octaviano. Challenges in heat network design optimization. *Energy*, 203:117688, 2020.

[201] S. Y. Wang and K. Tai. Graph representation for structural topology optimization using genetic algorithms. *Computers & Structures*, 82:1609–1622, 2004.

[202] C.Y. Wu and C.B. Zhang. Topology optimization of structures using ant colony optimization. In *Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO '05)*, pages 2251–2252, Washington DC, USA, 2005. ACM.

[203] A. Kaveh, B. Hassani, S. Shojaee, and S. M. Tavakkoli. Structural topology optimization using ant colony methodology. *Engineering Structures*, 30(9):2559–2565, 2008.

[204] G.C. Luh and C.Y. Lin. Structural topology optimization using ant colony optimization algorithm. *Applied Soft Computing*, 9:1343–1353, 2009.

[205] K.S. Yoo and S.Y. Han. Modified ant colony optimization for topology optimization of geometrically nonlinear structures. *International Journal of Precision Engineering and Manufacturing*, 15(4):679–687, 2014.

[206] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Milan, Italy, 1992.

[207] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed optimization by ant colonies. In *Proceedings of the First European Conference on Artificial Life (ECAL 91)*, pages 134–142, Paris, France, 1991. Elsevier.

[208] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.

[209] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NY, USA, 1999.

[210] J. Levine and F. Ducatelle. Ant colony optimization and local search for bin packing and cutting stock problems. *Journal of The Operational Research Society*, 55:705–716, 07 2004.

[211] P. Li, L. Wei, and D. Wu. An intelligently enhanced ant colony optimization algorithm for global path planning of mobile robots in engineering applications. *Sensors*, 25(5), 2025.

[212] G.E. Fainekos and K.C. Giannakoglou. Inverse design of airfoils based on a novel formulation of the ant colony optimization method. *Inverse Problems in Engineering*, 11(1):21–38, 2003.

[213] Marco Dorigo and Thomas Stutzle. *Ant Colony Optimization: Overview and Recent Advances*, pages 311–351. Springer International Publishing, Cham, 2019.

[214] T. Stutzle and H.H. Hoos. Max-min ant system. *Future Generation Computer Systems*, 16(8):889–914, 2000.

[215] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[216] B. Bullnheimer, R.F. Hartl, and C. Strauss. A new rank-based version of the ant system. *Central European Journal for Operations Research and Economics*, 7(1):25–38, 1999.

[217] T. Stützle and H. Hoos. Improvements on the ant-system: Introducing the max-min ant system. In *Artificial Neural Nets and Genetic Algorithms*, pages 245–249, Vienna, 1998. Springer Vienna.

[218] E.M. Papoutsi Kiachagias and K.C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering*, 23(2):255–299, 2016.

[219] N. Galanos. *Topology and shape optimization in fluid mechanics & conjugate heat transfer using continuous adjoint with consistent discretization*. PhD thesis, National Technical University of Athens, 2025.

[220] F.P. Incropera, D.P. DeWitt, T.L. Bergman, A.S. Lavine, et al. *Fundamentals of heat and mass transfer*, volume 6. Wiley New York, 1996.

[221] V. Gnielinski. New equations for heat and mass transfer in turbulent pipe and channel flow. *International Chemical Engineering*, 16(2):359–368, 1976.

[222] S.W Churchill. Friction factor equation spans all fluid-flow regimes. *Chemical Engineering*, 84(24):91–92, 1977.

[223] E. Salerno, A. Leonforte, M. Grespan, D. Angeli, and M.A. Corticelli. Cfd analysis of the thermal-hydraulic performance of traditional and alternative oils for transformer cooling. *Applied Sciences*, 14(21), 2024.

[224] A. Kulkarni, G. Brandes, A. Rahman, and S. Paul. A numerical model to evaluate the hvac power demand of electric vehicles. *IEEE Access*, 10:96239–96248, 2022.

[225] S. Chowdhury, L. Leitzel, M. Zima, M. Santacesaria, G. Titov, J. Lustbader, J. Rugh, J. Winkler, A. Khawaja, and M. Govindarajalu. Total thermal management of battery electric vehicles (bevs). In *CO2 Reduction for Transportation Systems Conference*. SAE International, 2018.

[226] B. Pulvirenti, G. Puccetti, and G. Semprini. Dynamic energy consumption modeling for hvac systems in electric vehicles. *Applied Sciences*, 15(7), 2025.

[227] L. Muratori, L. Peretto, G. Bottiglieri, F. Coiro, B. Pulvirenti, and R. Di Sante. Assessment of energy saving due to a flexible indoor air quality control. In *2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*, pages 175–180, 2021.

[228] F. Arpino, G. Cortellessa, G. Grossi, and H. Nagano. A eulerian-lagrangian approach for the non-isothermal and transient cfd analysis of the aerosol airborne dispersion in a car cabin. *Building and Environment*, 209:108648, 2022.

[229] F. Arpino, G. Grossi, G. Cortellessa, A. Mikszewski, L. Morawska, G. Buonanno, and L. Stabile. Risk of sars-cov-2 in a car cabin assessed through 3d cfd simulations. *Indoor Air*, 32(3):e13012, 2022.

[230] V. Mathai, A. Das, J.A. Bailey, and K. Breuer. Airflows inside passenger cars and implications for airborne disease transmission. *Science Advances*, 7(1):eabe0166, 2021.

[231] A.A. Sarhan, P. Naser, and J. Naser. Numerical study of when and who will get infected by coronavirus in passenger car. *Environmental Science and Pollution Research*, 29:57232–57247, 08 2022.

[232] D.E. Ramajo and S. Corzo. Airborne transmission risk in urban buses: a computational fluid dynamics study. *Aerosol and Air Quality Research*, 22(8):210334, 2022.

[233] X. Yang, C. Ou, H. Yang, L. Liu, T. Song, M. Kang, Lin H., and Hang J. Transmission of pathogen-laden expiratory droplets in a coach bus. *Journal of Hazardous Materials*, 397:122609, 2020.

[234] X. Xie, Y. Li, A.T.Y. Chwang, P.L. Ho, and W.H. Seto. How far droplets can move in indoor environments – revisiting the Wells evaporation–falling curve. *Indoor Air*, 17(3):211–225, 2007.

[235] W. F. Wells. On air-borne infection. study ii. droplets and droplet nuclei. *American Journal of Epidemiology*, 20(3):611–618, 11 1934.

[236] D. Parienta, L. Morawska, G.R. Johnson, Z.D. Ristovski, M. Hargreaves, K. Mengersen, S. Corbett, C.Y.H. Chao, Y. Li, and D. Katoshevski. Theoretical analysis of the motion and evaporation of exhaled respiratory droplet of mixed composition. *Journal of Aerosol Science*, 42(1):1–10, 2011.

[237] J. Redrow, S. Mao, I. Celik, J.A. Posada, and Z.G. Feng. Modeling the evaporation and dispersion of airborne sputum droplets expelled from a human cough. *Building and Environment*, 46(10):2042–2051, 2011.

[238] S. Balachandar, S. Zaleski, A. Soldati, G. Ahmadi, and L. Bourouiba. Host-to-host airborne transmission as a multiphase flow problem for science-based social distance guidelines. *International Journal of Multiphase Flow*, 132:103439, 2020.

[239] M. Cavazzuti and P. Tartarini. Transport and evaporation of exhaled respiratory droplets: An analytical model. *Physics of Fluids*, 35(10):103327, 10 2023.

[240] M. Cavazzuti, L. Campanelli, and P. Tartarini. Estimating infectious disease transmission risk: An analysis based on multi-scale modeling of respiratory droplet transport. *Physics of Fluids*, 37(2):023317, 2025.

[241] V. D'Alessandro, M. Falone, L. Giammichele, and R. Ricci. A multi-scale approach for modelling airborne transport of mucosalivary fluid. *Aerosol Sciences*, 12(23):12381, 2022.

[242] S. Mendez, W. Garcia, and A. Nicolas. From microscopic droplets to macroscopic crowds: Crossing the scales in models of short-range respiratory disease transmission, with application to covid-19. *Advanced Science*, 10(19):2205255, 2023.

[243] P.A. Mirzaei, M. Moshfeghi, H. Motamedi, Y. Sheikhnejad, and H. Bordbar. A simplified tempo-spatial model to predict airborne pathogen release risk in enclosed spaces: An eulerian-lagrangian cfd approach. *Building and Environment*, 207:108428, 2022.

[244] M.Y. Chong, H. An, P.C. Wang, C.B. Soh, S.C. Chien, E.R. Roch, J.J Koh, and S.C.M. Yu. Multiscale modelling and infection probability of sars-cov-2 transmission in a typical lecture theatre. *Building and Environment*, 269:112407, 2025.

[245] S. Mazumdar, Z. Long, and Q. Chen. A coupled computational fluid dynamics and analytical model to simulate airborne contaminant transmission in cabins. *Indoor and Built Environment*, 23(7):946–954, 2014.

[246] W.E. Ranz and W.R. Marshall. Evaporation from drops: Part 1. *Chemical Engineering Progress*, 48(3):141–146, 1952.

[247] W.E. Ranz and W.R. Marshall. Evaporation from drops: Part 2. *Chemical Engineering Progress*, 48(3):173–180, 1952.

[248] H.J. Holterman. *Kinetics and evaporation of water drops in air*. Number 2003-12 in IMAG rapport. IMAG, Netherlands, 2003.

[249] F.M. Raoult. Loi générale des tensions de vapeur des dissolvants. *Comptes rendus de l'Académie des sciences*, 104:1430–1433, 1887.

[250] J. Wei and Y. Li. Enhanced spread of expiratory droplets by turbulence in a cough jet. *Building and Environment*, 93:86–96, 2015.

[251] B. Wang, H. Wu, and X.F. Wan. Transport and fate of human expiratory droplets—a modeling approach. *Physics of Fluids*, 32(8):083307, 2020.

[252] T.G. Foat, B. Higgins, C. Abbs, T. Maishman, S. Coldrick, A. Kelsey, M.J. Ivings, S.T. Parker, and C.J. Noakes. Modelling the effect of temperature and relative humidity on exposure to SARS-CoV-2 in a mechanically ventilated room. *Indoor Air*, 32(11):e13146, 2022.

[253] M. Zhang, P. Shrestha, X. Liu, T. Turnaoglu, J. DeGraw, D. Schafer, and N. Love. Computational fluid dynamics simulation of SARS-CoV-2 aerosol dispersion inside a grocery store. *Building and Environment*, 209:108652, 2022.

[254] P.V. Nielsen. Computational fluid dynamics and room air movement. *Indoor Air*, 14(Suppl 7):134–143, 2004.

[255] ANSI/ASHRAE standard 55: thermal environmental conditions for human occupancy. Technical report, American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2013.

[256] L. Morawska, G.R. Johnson, Z.D. Ristovski, M. Hargreaves, K. Mengersen, S. Corbett, C.Y.H. Chao, Y. Li, and D. Katoshevski. Size distribution and sites of origin of droplets expelled from the human respiratory tract during expiratory activities. *Journal of Aerosol Science*, 40(3):256–269, 2009.

[257] J.W. Tang, A.D. Nicolle, C.A. Klettner, J. Pantelic, L. Wang, A.B. Suhaimi, A.Y.L. Tan, G.W.X. Ong, R. Su, C. Sekhar, D.D.W. Cheong, and K.W. Tham. Airflow dynamics of human jets: sneezing and breathing - potential sources of infectious aerosols. *PloS One*, 8(4):e59970, 2013.

[258] H. Li, F.Y. Leong, G. Xu, Z. Ge, C.W. Kang, and K.H. Lim. Dispersion of evaporating cough droplets in tropical outdoor environment. *Physics of Fluids*, 32(11):113301, 2020.

[259] J.K. Gupta, C.H. Lin, and Q. Chen. Flow dynamics and characterization of a cough. *Indoor Air*, 19(6):517–525, 2009.

[260] J.K. Gupta, C.H. Lin, and Q. Chen. Characterizing exhaled airflow from breathing and talking. *Indoor Air*, 20(1):31–39, 2010.

[261] S. Zhu, S. Kato, and J.H. Yang. Study on transport characteristics of saliva droplets produced by coughing in a calm indoor environment. *Building and Environment*, 41(12):1691–1702, 2006.

[262] W. Schroeder, K.M. Martin, and W.E. Lorensen. *The visualization toolkit, an object oriented approach to 3D graphics*. Kitware Inc., New York, 3rd edition, 2003.

[263] C.Y.H. Chao, M.P. Wan, L. Morawska, G.R. Johnson, Z.D. Ristovski, M. Hargreaves, K. Mengersen, S. Corbett, Y. Li, X. Xie, and D. Katoshevski. Characterization of expiration air jets and droplet size distributions immediately at the mouth opening. *Journal of Aerosol Science*, 40(2):122–133, 2009.

[264] L. Bourouiba. Fluid dynamics of respiratory infectious diseases. *Annual Review of Biomedical Engineering*, 23:547–577, 2021.

[265] S. Asadi, A.S. Wexler, C.D. Cappa, S. Barreda, N.M. Bouvier, and W.D. Ristenpart. Aerosol emission and superemission during human speech increase with voice loudness. *Scientific Reports*, 9:2348, 2019.

[266] R.H. Alford, J.A. Kasel, P.J. Gerone, and V. Knight. Human influenza resulting from aerosol inhalation. *Proceedings of the Society for Experimental Biology and Medicine*, 122(3):800–804, 1966.

[267] K.K.W. To, O.T.Y. Tsang, C.C.Y. Yip, K.H. Chan, T.C. Wu, J.M.C. Chan, W.S. Leung, T.S.H. Chik, C.Y.C. Choi, D.H. Kandamby, D.C. Lung, A.R. Tam, R.W.S. Poon, A.Y.F. Fung, I.F.N. Hung, V.C.C. Cheng, J.F.W. Chan, and K.Y. Yuen. Consistent detection of 2019 novel coronavirus in saliva. *Clinical Infectious Diseases*, 71(15):841–843, 2020.

[268] J.M. Kolinski and T.M. Schneider. Superspreading events suggest aerosol transmission of SARS-CoV-2 by accumulation in enclosed spaces. *Physical Review E*, 103(3):033109, 2021.

[269] E.C. Riley, G. Murphy, and R.L. Riley. Airborne spread of measles in a suburban elementary school. *American Journal of Epidemiology*, 107(5):421–432, 1978.

[270] N. Van Doremalen, T. Bushmaker, D.H. Morris, M.G. Holbrook, A. Gamble, B.N. Williamson, A. Tamin, J.L. Harcourt, N.J. Thornburg, S.I. Gerber, J.O. Lloyd-Smith, E. de Wit, and V.J. Munster. Aerosol and surface stability of SARS-CoV-2 as compared with SARS-CoV-1. *New England Journal of Medicine*, 382(16):1564–1567, 2020.

[271] M. Cavazzuti and P. Tartarini. Statistical analysis of infectious disease transmission risk based on exhaled respiratory droplet trajectory distribution. *Physics of Fluids*, 36(6):063341, 2024.

[272] E. Galligani. Dispense del corso di metodi numerici per l'ingegneria, 2019.