POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Energetica e Nucleare

Tesi di Laurea Magistrale

2D CFD modelling of a liquid metal box type divertor

Relatori
prof. Laura SAVOLDI
prof. Antonio CAMMI

Correlatori
prof. Fabio SUBBA
prof. Roberto ZANINO
dott. Stefano LORENZI
ing. Giuseppe Francesco NALLO

Candidato
Andrea IOVENITTI

ANNO ACCADEMICO 2017-2018

A mio padre

Abstract

One of the major issues concerning tokamak fusion reactors is to deal with the heat and particle flux coming from the core plasma and impinging on specific plasma facing components, which are called divertors. Divertors must withstand high thermal and particle loads, that can cause unwanted phenomena such as surface erosion or melting. Therefore, a reliable strategy for power exhaust is needed.

Among the different proposals, the use of liquid metal (LM) for coating the divertor is one of the most interesting option. In particular, the present thesis deals with a box type LM divertor. This divertor is made of an evaporation chamber (EC), where a LM pool is located, and a differential chamber (DC), that is required in order to reduce as much as possible the probability for a metal particle to reach the core plasma, causing its contamination.

Preliminary studies aimed at describing the physics inside a box pool type LM divertor have been proposed in literature. However, they are under development and need to be further investigated. In particular, the engineering model this thesis refers to exploits a 0D model to compute the thermodynamic variables of the LM.

The scope of the present work, that is divided in two parts, is to perform a 2D analysis aiming to provide a better evaluation of LM thermal phase change and differential pumping of metal vapor, thereby improving the predictions of the currently available model.

In the first part, a Computational Fluid Dynamic (CFD) model, aimed of evaluating thermal phase change, is developed and applied to the LM divertor. The EC is in fact expected to have such conditions that the metal can be treated as a continuum.

In the second part, a Direct Simulation Monte Carlo analysis is performed to investigate the differential pumping. Indeed, the DC is found to be in a transitional regime, so a molecular approach is needed.

Contents

Chapter 1

Introduction

During the last decades, the total amount of electrical energy consumed by the world population has dramatically increased; as a consequence, alternatives to fossil fuels have been studied, in order to satisfy the energy demand in a sustainable way [1]. In this context, nuclear fusion has nowadays become one of the most interesting options. As it is explained in [2], the European Union is currently providing support, under the Horizon 2020 Program, for the realization of the International Thermonuclear Experimental Reactor (ITER), whose goal is to show the effective applicability of this new energy source. The first reactor to demonstrate the economic competitiveness of fusion electricity, DEMO, is also undergoing a conceptual design in the framework of the EuroFusion consortium.

Both ITER and DEMO are tokamaks (*toroidal'naya kamera s magnitnymi katushkami* — toroidal chamber with magnetic coils) fed by deuterium and tritium, which coexist in form of plasma. Their fusion reaction produces an α particle having an energy of 3.5 MeV, which is used to heat up the system (the reaction is thermally sustained), and a neutron at 14.1 MeV, that is employed to produce electricity. The plasma is confined inside the main chamber by means of a magnetic field which is obtained by combining three different components (see [3] for details):

- *toroidal*, whose role is to provide an higher stability. It is produced by many Toroidal Field coils (18 in the case of ITER);
- *poloidal*, aimed at providing confinement. In a tokamak, this component is obtained inductively: a Central Solenoid produces a magnetic field (Ampere's law) which determines a toroidal electric field (Faraday's law), which is responsible for the production of the poloidal magnetic field. Therefore, the poloidal magnetic field is time dependent (conversely, in a Stellarator it is constant and this is the result of a complex geometry);
- *vertical*, which balances those forces coming from the toroidal geometry of the machine. It is realised by the Poloidal Field coils.

Figure 1.1 shows the typical structure of a tokamak. The pink zone represents the main chamber; in particular, this scheme refers to ITER. The three different

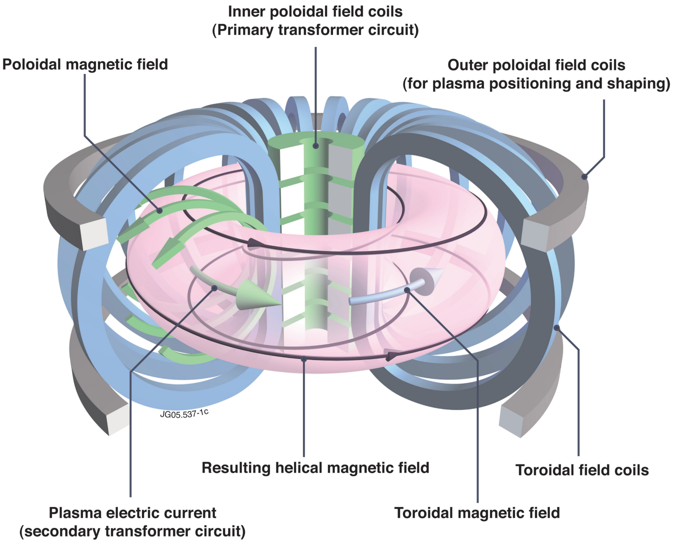

Figure 1.1: Schematic structure of a tokamak. Adapted from [4]

type of magnets can also be seen. Ideally, no other magnets are needed; however, in reality there are also some additional coils, which correct engineering errors, such as magnetic ripples due to the discontinuities of the TF coils discontinuities.

Since the magnetic confinement is not perfect, two different zones can be distinguished inside the main chamber:

1. the core plasma, where the magnetic field lines are closed on themselves;
2. the Scrape-Off Layer (SOL), where they intersect the solid walls before closing (see Figure 1.2).

Density gradients exist in the core (larger density in the center, lower density at the edge). As a consequence, As a consequence, through Coulomb collisions, particles can diffuse from the core towards the SOL.

After these particles have crossed the *separatrix* (that is the last closed magnetic surface), they enter the SOL, where parallel-to-$\vec{B}$ transport dominates over radial diffusion of particles. Due to this fact, particles will only have time for a small displacement in radial direction before hitting the divertor target with a speed equal to the local sound speed (ref. [7]). Divertor targets (see Figure 1.2) are for this reason the most severely loaded among the plasma-facing components (PFCs) High energy particles impinging on the divertor will cause:

- *sputtering*: the particle flux can mechanically erode them. This has two main consequences; on the one hand, sputtered particles might reach the core plasma and contaminate it, that is already unstable. Indeed, Bremsstrahlung radiation due to Coulomb collisions determines energy losses which can

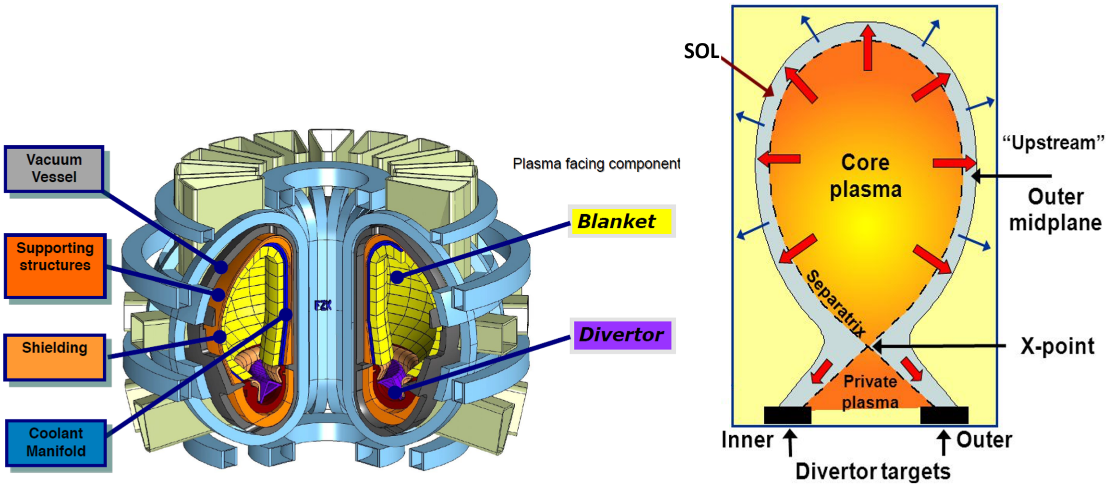

Figure 1.2: On the left, design of DEMO fusion reactor (adapted from [5]. On the right, schematic representation of the plasma physics inside the MC (adapted from [6]).

switch the reaction chain off. A key parameter quantifying this effects is the effective charge of all the ions Z_{eff}; the higher Z_{eff}, the higher the Bremsstrahlung loss. Clearly, the presence of sputtered particles (which are heavier than hydrogen ions) let the effective charge increase (which is not desirable). On the other hand, sputtering produces wall damage which may lead to the need of frequently replacing PFCs;

- *heat loads*: although particles impinging these PFCs are much colder than those in the core plasma (which are at 1-15 keV), the resulting heat flux can still be large enough to melt solid walls, even when made of materials such as tungsten. Indeed, it can be shown that the energy transferred by each particle is around three times (see [7]) that associated to its temperature, which is in the range (1,100) eV. This results in estimates of tens of $\mathrm{MW/m^2}$ for ITER.

It is beyond the scope of this work to deal with plasma-wall interactions, for which we refer to specialized literature [7]; nevertheless,in the following a rough estimate is provided in rder to indicate the severity of the phenomena. Reference values for ITER are taken from [6] and reported in Table 1.1.

It seems relevant to highlight that, when performing the design of the divertors, the knowledge of the power width λ_q is fundamental. This quantity represents the characteristic length of the exponential decay that power feels in the radial direction; therefore, λ_q is desirable to be as large as possible. Unfortunately, it has been recently found that its value can be even in the range of 1-2 mm (a non conservative value of 5 mm is here assumed); this is a consequence of the aforementioned parallel and perpendicular to $\vec{B}$ transport characterizing the SOL. Few scalings are available for λ_q (ref. [8] and [9]), which have been heuristically derived and then

Table 1.1: ITER data

Quantity	Symbol	Value	Unit
Major radius	R	5.3	m
Poloidal angle	β	25	°
Power width	λ_q	5.0	mm
Magnetic ratio upstream	$(B_\theta/B)_u$	0.33	–
Magnetic ratio target	$(B_\theta/B)_t$	0.11	–

benchmarked against present experimental data. A reference value of 5 mm is here assumed.

The area available for exhausting the particle flux coming from the SOL can be evaluated as:

$$Area \sim \frac{2\pi R_t \cdot \lambda_q \cdot (B_\theta/B)_u}{sin(\beta) \cdot (B_\theta/B)_t} \sim 1.2m^2 \tag{1.1}$$

Since the heat loads impinging the divertors are very high (expected power in ITER is $\sim$ 100 MW) while the affected area is small, huge heat fluxes of several tens of MW/m^2 result. Moreover, in large machines, the total fusion power (MW) will increase, whereas the wetted area will only slightly increase due to the fact that λ_q shows - unfortunately - no size scaling. As a consequence, although they are made of tungsten, it is fundamental to provide a reliable strategy for power exhaust. The idea of liquid metals, discussed in the following section, seems to be a valid solution.

1.1 Liquid metal divertors

The use of liquid metals to sustain the extreme heat and particle flux impinging on the divertors dates back to the 90s. Many earlier ideas which had been suggested to solve the problem (see [10]) have turned out not to be acceptable, because of the difficulty of an actual reactor implementation; that is why Golubchinov et al. [11] proposed to use a liquid metal divertor. In the proposed concept, the liquid metal (LM) is held in place by a Capillary Porous Structure (CPS), which is basically the same as they are employed in heat pipes (ref. [12]). This kind of system is self-sustaining and self-regulating, because the amount of liquid replacing the evaporated one is pressure dependent; therefore, the higher the evaporation rate, the higher the pressure difference and so the replacement rate. This idea was further investigated by Evtikhin et al. [13], which gave a first estimation of the capillary pressure difference and performed some experimental studies. As a consequence of LM evaporation, the target is cooled (evaporated mass carries away latent heat) and the plasma is cooled in turn due to energy loss associated to ionization of the evaporated neutral atoms. On the one hand, this implies reducing temperature gradients inside the solid structures and so thermal stresses; on the other hand, surface erosion can be almost absent. In addition, very serious potential damages such as those resulting from Edge Localized Modes, as well as neutron irradiation, could be significantly diminished.

In 2009, Nagayama proposed four different concepts of LM divertor systems [14]. They all have an evaporation chamber (EC) which is separated from the main chamber (MC) through another camera, that is called differential chamber DC. The role of the latter is to realize differential pumping between the EC and the MC, thereby reducing the intrinsic impurity flux which exits the system and which might contaminate the core plasma. It is desirable that apertures between adjacent chambers are small, so that the probability for a splashed particle to reach the core plasma is reduced. However, it should be also pointed out that nozzles have to be at least larger than the aforementioned power width λ_q, that is the length scale of the radial exponential decay of power. Moreover, when the magnetic field feels small perturbations, the striking point can move in such a way that the particle flux can impinge on the walls, causing both sputtering and local melting. An optimization process is therefore needed when designing these openings.

Figure 1.3 schematically shows the proposed systems; two of them are pool types, while the other two have a wick (i.e. a CPS). Nagayama explains that the pool schemes have many weak points that can be avoided adopting CPS divertors, thus confirming what Golubchinov had already suggested.

Ono et al. [15] proposed in 2013 a different type of divertor, called Radiative Liquid Lithium Divertor RLLD: the vapour metal resulting from evaporation is rapidly ionized by the plasma and can strongly radiate. This process can let heat loads be distributed much more isotropically, thereby facilitating the heat removal. According to this proposal, liquid lithium is introduced in the upper part of the divertors and flows down thanks to gravity (this let MagnetoHydroDynamics - MHD - not to be relevant). Porous structures can be adopted to reach a more uniform fluid distribution. The RLLD concept was further developed, leading to a slightly different idea, that is the Active RLLD (ARLLD) [16]; liquid lithium is actively injected from the divertor walls in such a way that it can be rapidly delivered to the plasma. In this way, the radiation processes could happen quite far from the target plate, implying that radiated power is redistributed on a larger area, significantly reducing the peak heat fluxes.

Figure 1.3 shows a simplified schemes of the RLLD and ARLLD. It appears relevant to point it out that the radiation process can dissipate a much larger (up to three order of magnitudes) amount of the parallel heat flux than evaporation processes. Therefore, with respect to Nagayama idea, an additional power dissipation channel has been included in the picture.

In 2015, Goldston et al. [8] combined the ideas of Nagayama and Ono, proposing a series of vapor boxes to reduce as much as possible the amount of impurities

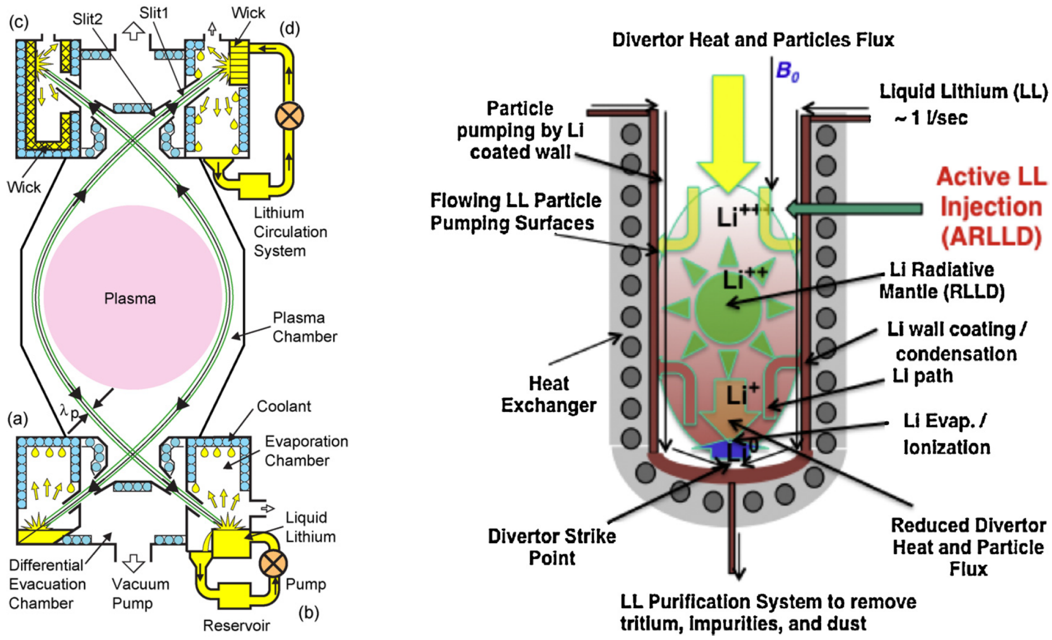

Figure 1.3: On the left, schematic view of the four liquid metal divertor system proposed by Nagayama: (a) lithium pool; (b) lithium pool and circulation pump; (c) wick; (d) wick and circulation pump (adapted from [14]). On the right, a simplified schematic of RLLD chamber (adapted from [15]).

reaching the MC from the EC. In Nagayama's idea there is a key issue concerning the evaporated metal: it can be ionized, returning immediately to the divertors surfaces because of the plasma flux coming from the SOL. If this happens, the cooling effect due to phase change is substantially weakened. Ono pointed out that interactions between evaporated metal and incoming plasma can be beneficial, since they reduce the parallel heat flux by means of ionization and radiation processes.In this way, the combination of evaporation and irradiation could be absolutely useful.

When passing through the apertures, vapor can reach sonic conditions. To verify that, inviscid Navier-Stokes calculations, optimized to resolve shocks, have been performed in [17]; when setting up the simulations, phantom boundary cells of Maxwellian vapor at the wall temperature are used to simulate the effect of evaporation and condensation at the walls. Simulations have shown that the configuration with successive boxes is able to confine vapor mostly in the chambers closest to the pool.

Goldston's vapor box divertor seems to be the most interesting option among those discussed. Because of this, a first engineering model of a box pool-type divertor has been developed [18]. That model is the object of the next paragraph.

1.1.1 An engineering model of a box type divertor

According to [18], there are three fundamental pieces of physics that must be taken into account when dealing with a box type divertor. A model has recently been proposed to deal with this physics in a self-consistent way.

1. the impinging plasma flux is the driving phenomenon. As already stated, particles diffuse from the core plasma to the SOL, where they flow advectively towards the divertors along the magnetic field lines, eventually reaching sound speed at targets. It is mandatory to compute the flux distribution and, consequently, the heat loads that must be withstood. This estimations is performed through a 1D SOL plasma model, which requires as an input upstream plasma conditions (parallel heat flux and density). The model evaluates the temperature, density, velocity and heat flux profiles, as well as the power repartition among conduction, convection and radiation;

2. the plasma power is transferred to the liquid metal which partially evaporates. As already mentioned, the resulting vapor can strongly radiate and this can be very beneficial in order to reduce the heat flux to the divertor plates. A 0D model developed by Nallo et al. [19] is used to evaluate the thermodynamic state of the LM and the vapor in the chambers, the evaporation at the strike point and the condensation at the solid walls. The required input are the plasma power reaching the pool - evaluated by the aforementioned 1D SOL model - and the wall temperatures;

3. a 2D model is needed to take into account the thermal conduction in the structures. Boundary conditions are given by heat loads - both due to radiation associated to plasma-vapor interactions and to condensation heat loads - and by active cooling of the walls.

An iterative approach is proposed: upstream plasma conditions are set and the 1D SOL model is solved. Once this calculation reaches convergence, the power distribution is obtained and used as an input to the 0D thermodynamic model. Finally, the 2D conductive model is solved, because it requires as input both the fraction of radiative power which is transferred to the structures and the heat transfer with the liquid metal. The three models are then run again iteratively until outer convergence is reached. A schematic representation of this coupling is shown in Figure 1.4.

1.1.2 Choice of liquid metal

Lithium was the first liquid metal to be considered for the purpose of covering the divertor walls ([11] and [13]); indeed, it has a lot a attractive features:

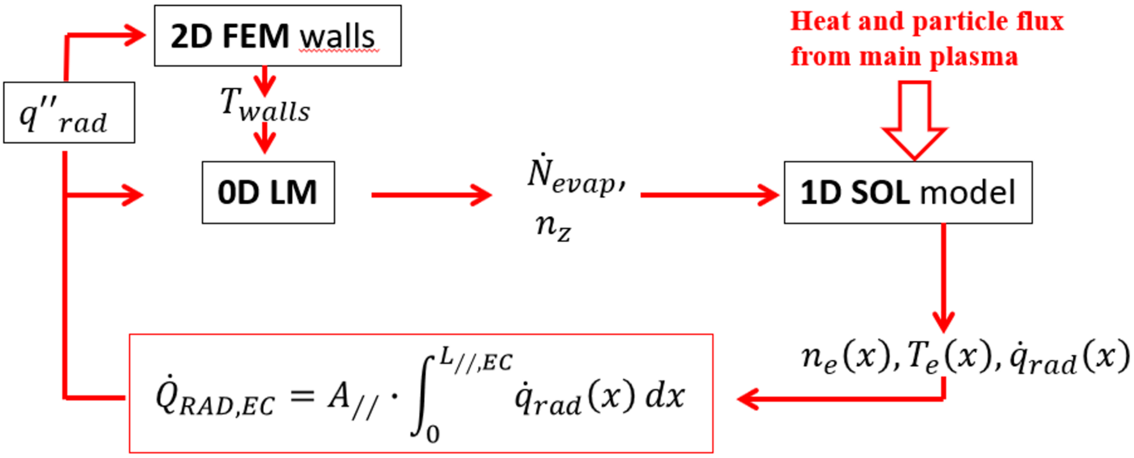

Figure 1.4: Schematic of the coupling strategy between the modules of the code (adapted from [18]).

- low atomic number ($Z = 3$): if lithium reaches the core plasma, the contribution to the effective charge of all the ions should be low even with a significant amount of Li;
- high latent heat ($\sim 1.956e7\ J/kg$ at $T \sim 800 - 1000\ K$): this is beneficial for the design purposes, since it allows a significant amount of power be dissipated through phase change, instead of being transferred to the solid structures;
- experimental studies [15] have shown that the use of lithium (see also Figure 1.5):
  - broadens the electron temperature profile and, consequently, improves the energy confinement time;
  - decreases the D_α signal, that is a measure giving indications on the recycling of deuterium gas. In other words, this means that lithium behaves as a passive pump for deuterium. ELMs effects are also dramatically reduced thanks to this "low recycling" regime;
  - reduces, through chemical reactions, the amount of residual gases which are typical for vacuum systems (i.e. oxigen) [20]

Moreover, techniques associated to blanket cooling (e.g. Water Cooled Lithium Lead) might also be exploited, since they are similar to LM divertors in terms of material compatibility.

At the beginning, interest for Li divertor coatings was associated to the possibility to operate in the afromentioned high performance, low recycling regime. However, Nygren et al. [21] put a base limit for low and high recycling at $\sim 450\ °C$; at

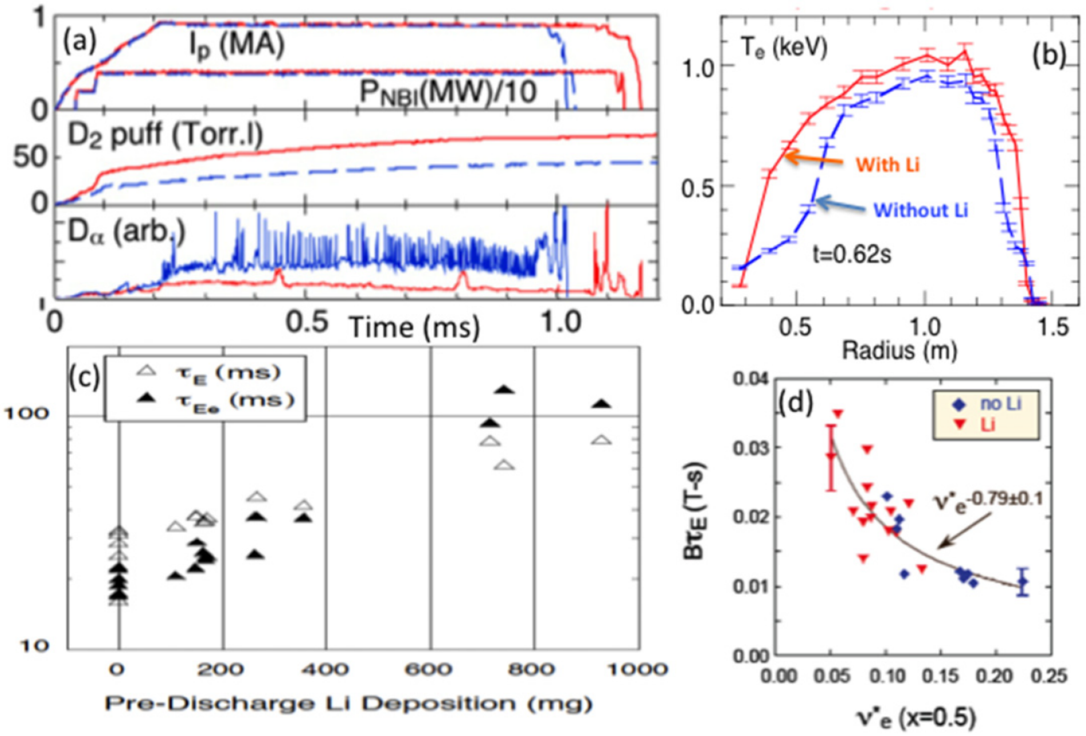

Figure 1.5: Plasma performances improvements with Li. (a) Plasma discharge traces with and without Li in NSTX. Red traces are with Li and blue-dashed are without Li. (b) Electron temperature profiles with Li (red) and without Li (blue-dashed). (c) Total and electron energy confinement time (τ_E and $\tau_{E,e}$, respectively) versus pre-discharge Li evaporation amount. (d) Normalized energy condinement time versus electron collisionality and without Li as labelled (adapted from [15]).

higher temperatures, LiD decomposition starts being relevant for slow-flow divertors (as the box pool-type - slow motion let MHD losses be neglected). This leads to a fundamental distinction among Li-based divertor concept:

- fast-flowing concepts (ref. [22]), where a low temperature is maintained by Li movement and therefore low recycling is reached;
- slow-flow concepts, such as CPS, where Li is moving at very low speeds and it will therefore reach high temperature, thus losing the advantages associated to a low recycling regime.

The concept explored in the work is part of the latter group.

Tin is another possible candidate to be used for liquid metal divertors, even if the interest on it is quite recent and detailed studies are ongoing ([23]). Some relevant features of this fluid are:

- very high latent heat (more than 30 times larger than that of Li): the evoporation/condensation rate is very small, so a divertor without a box structure could in principle be employed ([21] and [18]);
- high atomic number: plasma cooling by metal plasma interactions is an effective mechanism. However, if tin reaches the core plasma, this would be problematic;
- less chemically aggressive than lithium: because of this, corrosion compatibility is better than in the case of lithium ([21])

The best choice for a vapor box divertor is Li, according to the conclusions of [18]. This will therefore be the only LM considered in this work.

1.2 Aim of the work

Although many fundamental aspects of the physics of a box pool-type divertor are caught by the multiphysic model described in section 1.1.1, still some improvements can be done. In particular, the use of a 0D model does not allow to estimate thermal hotspots, because only average quantities are computed. The effect of the specific geometry is another feature which is completely lost by a 0D approach.

The aim of the present thesis work is therefore to perform a 2D (toroidally symmetric) analysis aimed at providing a more accurate evaluating the distributions of pressure, temperature and density inside each chamber of the box type divertor, thereby improving the predictions of the currently available model. In particular, the following aspects will be key features of the analysis:

- evaporation/condensation of the liquid metal;
- differential pumping of vapor;

Indeed, it is thought that the present 0D model might be reasonably adequate for fast estimates which are required during the process, but that more detailed modeling is required to catch details such as hot spots and the distribution of the condensed phase.

1.3 Conclusions

Nowadays, nuclear fusion represents one of the most promising and interesting option to face the world electrical energy problem, as it is confirmed by the great amount of research activities and financial investments. Among the different strategies, magnetic confinement seems to be the most favorable one. However, relatively hot plasma can reach the walls, determining huge particle and heat loads.

Divertors are the components which have to be properly designed, in order to guarantee a safe and reliable way to address heat removal. Among the different options, liquid metal divertors are believed to be promising and many studies are currently ongoing. In particular lithium seems to be the most promising candidate is a box type LM divertor is concerned. Tin appears instead to be a better choice when dealing with CPS.

An engineering model of a box type divertor has already been proposed. In particular, the thermodynamic state of the liquid metal is evaluated by a 0D model. The aim of the present thesis is to perform a 2D analysis of a Li vapor box divertor, especially focused on thermal phase change an on differential pumping.

The main results derived have been presented at the *International Conference on Plasma Surface Interactions in controlled fusion devices,* Princeton University, 17-22 June 2018.

References

[1] EFDA. «World Energy Outlook 2017». 2017.

[2] «A roadmap to the realisation of fusion energy». 2012.

[3] J.P. Friedberg. «Plasma Physics and Fusion Energy». 2007.

[4] G.F. Nallo. «Liquid metal divertor modeling for future fusion reactors». 2016.

[5] «Blanket technology foa the fusion reactor». 2015.

[6] R. Pitts. «Edge physics and plasma-wall interactions». 2011.

[7] P.C. Stangeby. «The plasma boundary of magnetic fusion devices». 2000.

[8] R.J. Goldston et al. «The lithium vapor box divertor». In: *Royal Swedish Academy of Sciences* (2015).

[9] T. Eich et al. «Scaling of the tokamak near the scrape-off layer H-mode power width and implications for ITER». In: *IAEA conference in Vienna* (2013).

[10] «ITER Concept Definition, ITER-1». 1988.

[11] L.G. Golubchikov et al. «Development of a liquid-metal fusion reactor divertor with a capillary-pore system». In: *Journal of Nuclear Materials* (1996).

[12] A. Foghri. «Heat pipe and technology». 1995.

[13] V.A. Evtikhin et al. «Calculation and experimental investigation of fusion reactor divertor plate and first wall protection by capillary-pore systems with lithium». In: *Journal of Nuclear Materials* (1999).

[14] Y. Nagayama. «Liquid lithium divertor system for fusion reactor». In: *Fusion Engineering and Design* (2009).

[15] M. Ono et al. «Recent progress in the NSTX/NSTX-U lithium programme and prospects for reactor-relevant liquid-lithium based divertor development». In: *Nuclear Fusion* (2013).

[16] M. Ono et al. «Active radiative liquid lithium divertor concept». In: *Fusion Engineering and Design* (2014).

[17] R.J. Goldston et al. «Recent advances towards a lithium vapor box divertor». In: *Nuclear Materials and Energy* (2017).

[18] G.F. Nallo et al. «Development of a self consistent model for determining engineering parameters of a closed LM divertor». 2017.

[19] G.F. Nallo et al. «Modeling the lithium loop in a liquid metal pool-type divertor». In: *Fusion Engineering and Design* (2017).

[20] M.A. Jaworski et al. «Upgrades towards high-heat flux, liquid lithium plasma-facing components in the NSTX-U». In: *Fusion Engineering and Design* (2016).

[21] R.E Nygren et al. «Liquid surfaces for fusion plasma facing components - A critical review. Part I: physics and PSI». In: *Nuclear Materials and Energy* (2016).

[22] D.N Ruzic et al. «Flowing liquid lithium plasma facing components - Physics, technology and system analysis of the limit system». In: *Nuclear Materials and Energy* (2017).

[23] G.G. Van Enden et al. «Self-regulated plasma heat flux mitigation due to liquid Sn vapor shielding». 2016.

Part I

Assessment of the feasibility of a numerical evaluation of the liquid metal condensation on the box walls through Computational Fluid Dynamics

Chapter 1

Introduction: Part I

As it has been discussed in the previous chapter, in a box type divertor many different physical phenomena must be accounted for (e.g. boiling/evaporation, condensation, differential pumping, irradiation). Among them, the interactions between the vapor metal and the plasma which can lead to plasma entrainment represent a peculiar feature characterizing plasma facing components such as divertors. Therefore, the problem is intrinsically multiphysics and involve both the study of the metal behavior, the conduction at the solid walls and the inter-particles reactions. Software such as STAR-CCM+ (that is mainly focused on CFD) might not be adequate to catch all the most important pieces of the physics.

Furthermore, the engineering model described in section 1.1.1 takes advantage of different modules, each being used to model a specific part of the physics that is studied (see Figure 1.4). As a consequence, it might be required to couple different codes and this could be troublesome when dealing with commercial tools such as ANSYS. As a matter of fact, the 2D thermal conduction at the walls is modeled using the open source code FreeFem++.

Because of all these reasons, OpenFOAM 5.0, which uses the Finite Volume Method, is chosen to be used within this work. Being open-source, this tool is highly flexible, parallelizable and gives the chance to freely modify equations, introducing new formulations which might be needed when simulating a specific problem such as thermal phase change, for which there are no available solvers which have been already verified and validated. Moreover, the open community of users and developers which actively participate to the continuous improvement of the software is certainly fundamental when testing new solvers or models («given enough eyeballs, all bugs are shallow» - Linus's law).

1.1 State of art

Thermal phase change is believed to be the most important aspect to be modeled, since it is the driver that produces the Li vapor that is inside the divertor chambers. When dealing with this kind of phenomena, it is very easy to to find in literature a lot of experimental correlations which refer to different situations. Nukiyama firstly studied pool boiling, identifying different regimes and defining the concept of critical heat flux [1]. Later on, convective boiling was deeply investigated and many correlations have been derived [2]; for example, Jens-Lottes or McAdams formulations are used to determine the fully developed boiling condition, while Tong or Bowring deal with critical heat fluxes. On the contrary, CFD simulations dealing with phase change still represents a developing field.

In OpenFOAM, available multiphase solvers can be divided in two groups: Euler-Euler and Volume Of Fluid [3].

1. In the *Euler-Euler* method, a set of conservation equations (mass, momentum and energy) is solved for each phase (two fluid approach). Liquid and vapor are miscible and can interpenetrate each other. As a consequence, when phenomena such as dispersions, bubbly and/or slug flows and similar have to be investigated, the use of an Euler-Euler solver is mandatory [4]. From a mathematical point of view, the interface is modeled through coupling terms between the two set of equations; therefore, many aspects of the physics are captured. However, a lot of input parameters for the interface coupling models, such as those related to the interface (mass transfer, heat transfer, drag, surface tension etc.), are required because of the great amount of details that is provided. Setting up a simulation might be very difficult and its computational cost is definitely not negligible, since a lot of equations has to be solved for each cell volume of the computational domain;

2. the *Volume Of Fluid* family of solvers computes only one set of conservation equation (one fluid approach), assuming that at the interface between two phases there is no pressure jump. The velocity field is obtained from the phases velocities, each being multiplied by its volume fraction. Finally, a volume fraction conservation equation is solved to track the interface and, consequently, the volume of each phase. It is fundamental to stress that in OpenFOAM the VOF method does not include any procedure to reconstruct the interface [5]. The latter is just compressed in a suitable way and an algebraic method is applied when solving the volume fraction equation (see chapter 3).

   It is much easier to use VOF solvers than Euler-Euler. Moreover, the computational cost is more affordable. The negative aspect of these solvers is instead the impossibility to simulate bubbling flows or situations where dispersion phenomena of one phase into another one are relevant.

In OpenFOAM, many attempts to study phase change problems through CFD have been performed adopting a one fluid approach. In particular, there is an existing solver which is for phase-change. However, the physical models that are implemented only take into account cavitating phenomena (i.e. pressure-driven phase change). Moreover, it must be noticed that the energy equation is not implemented. Therefore, programming is needed.

To the author's knowledge, the first attempt to study direct contact condensation in OpenFOAM was performed in 2010 [6]. Two additional models were included: AlbaNovaInterface and AlbaNovaCombustion. However, both of them are considered not to be adequate for modeling condensation in an acceptable way. As a matter of fact, some weak points of AlbaNovaInterface are:

- the heat transfer coefficient is computed using the relation Nusselt derived for laminar film condensation. This formulation is clearly not general;
- the procedure to compute the interfacial area is too simplistic and inaccurate.

Furthermore, no energy equation was implemented.

In 2011, at Chalmers, another attempt to model thermal phase change was performed [7]. The major improvements are the addition of the temperature equation and the implementation of the Auguste-Roche-Magnus formula to establish a temperature dependence of the saturated pressure. However:

- the temperature equation does not have any source/sink terms related to phase change;
- no phase change models are implemented, but those given by default and used for cavitation are used.

As a consequence, the developed code is believed not to be reliable by the author.

In 2012 Kunkelmann proposed for the first time to model thermal phase change through reconstructing algorithms (Piecewise Linear Interface Calculation and contour based reconstruction were suggested) to capture the interface[5]; these algorithms are also used to evaluate the evaporating mass flux. At the same time, another formulation, based on the work of Hardt and Wondra [8], was also proposed as an alternative; in particular, it was found that to achieve the same level of accuracy, the use of reconstructing algorithms allows the user to adopt a mesh that is four times coarser than the one needed when computing the evaporating flux through the model of Hard and Wondra. This clearly goes in the direction of speeding the simulation. However, the use of reconstructing algorithms also implies a larger number of calculations that has to be performed and this has a negative impact in terms of computational cost. Because of this trade off and because the implementation of Hardt and Wondra is straightforward, Kunkelmann decided to adopt the second approach in his work.

Bruno Santos further modified Kunkelmann's code, making some slightly different changes that are of numerical nature (the resulting solver is available online [9]). However, after having tested the code on reference problems (1D Stefan evaporation and 1D Stefan condensation), it came out that:

- while the solver works nicely for evaporation, condensation is an issue;
- mass conservation is not verified.

Nevertheless, it is believed that Kunkelmann's work is a relevant starting point to understand the main physical and numerical issues associated to thermal phase change modeling.

Guedon also tried to face the problem of two phase heat and mass transfer modeling [10]. He followed the same line of thoughts of Kunkelmann and implemented new interfacial curvature methods. He also performed many verification tests, which are not only related to simple cases such as 1D Stefan problem, but also on rising bubbles or nucleate pool boiling. Despite this, mass conservation issues are not addressed in his work.

While both Kunkelmann and Guedon have chosen to implement the temperature energy equation, Nabil and Rattner preferred to use internal energy [11]. This is clearly the best choice, since phase change is under study. The implementation of the code is anyhow questionable. Indeed, in OpenFOAM, the use of enthalpy/internal energy variables is strictly related to the use of an informatic class that is called "thermophysicalModels", whose main role is to model the thermodynamic state of the fluid under analysis through the particular equation of state selected by the user. Since the reference solver from which Nabil and Rattner developed their code deals with incompressible fluids, another class, called "transportModels" is used by default. This class is only used for modeling transport and no modifications have been made to include thermophysical models. The approach followed by Nabil and Rattner seems therefore contradictory is this sense.

As a matter of fact, in the verification test shown in [12] the working vapor has non-physically modified thermophysical properties, which are used to smooth the sharp variations existing at the interface: this is very helpful when running the code, but it also means the fluid under simulation is wrong. Moreover, it seems that volume are conserved, while masses are not. The relative error is anyway small because the total amount of mass at time t=0 s is large (water is heated and boiling occurs). A condensating case study should be useful to further investigate this issue, but it is not provided by the authors.

1.2 Conclusions

The 2D model of a box type divertor is intrinsically a multiphysics problem, which couple different phenomena, such as conduction at the solid walls, thermal fluid dynamics of the metal, vapor irradiation, nuclear reactions between the plasma and the vapor, which can be entrained and recombined (especially near the pool). Therefore, it may be required to couple different codes, each studying a relevant piece of the physics.

The open source software OpenFOAM is thereby chosen, since it gives the chance to freely modify equations in order to model what is desirable.

However, it does not have any developed application which deals with thermal phase change. After a deep bibliographic study, it was concluded that it is mandatory to try to implement a reliable CFD code to deal with boiling/condensation.

References

[1] F.P. Incropera et al. *Funfamentals of heat and mass transfer*. 1996.

[2] J.G. Collier et al. *Convective boiling and condensation*. 1994.

[3] *OpenFOAM 5.0 user guide*. 2017.

[4] *Fluent 6.0 user's guide*. 2001.

[5] C. Kunkelmann. «Numerical modeling and investigation of boiling phenomena». 2016.

[6] R. Thiele. «Modeling od direct contact condensation with OpenFOAM». 2010.

[7] M. Andersen. «A interphaseChageFoam tutorial». 2010.

[8] F. Wondra S. Hardt. «Evaporation model for interfacial flows based on a continuum-fields representation of the source terms». In: *Journal of computational physics* (2008).

[9] *https://github.com/wyldckat/evapVOFHardt*. 2016.

[10] G. Guedon. «Two-phase heat and mass transfer modeling: flexible numerical methods for energy engineering analyses». 2013.

[11] M. Nabil et al. «interThermalPhaseChangeFoam-A framework for two phase flow simulations with thermally driven phase change». 2016.

[12] A.S. Rattner et al. «Simulation of Taylor flow evaporation for bubble-pump applications». In: *International journal of heat and mass transfer* (2018).

Chapter 2

Governing equations

The physical model which is developed in the present thesis is mostly based on Nima Samkhaniani's work, that is believed to be one of the best available in literature ([1] and [2]). The approach is based on VOF methods and conservation equations are written for the one fluid (i.e. the mixture between the liquid and vapor phases).

After stating the assumptions of the model, each equation is discussed and the single terms are analyzed in detail; then, the physical treatment of the volumetric phase change density rate is addressed.

2.1 Assumptions

The following assumptions are made:

1. *fluid as a continuum*: since a CFD analysis is going to be performed, both fluids must be continua inside the computational domain. This assumption is physically true for the EC of the liquid metal divertor, while it is not verified for the DC (see [3] for details). As a consequence, boundary conditions of the liquid metal divertor will always refer to the EC, even if the geometry is such that both chambers are taken into account;

2. *dispersion and/or interpenetration*: these phenomena between different phases are not relevant for the purposes of this work and therefore are neglected, as well as chemical reactions (indeed, the plasma heat flux coming from the SOL could react with the vapor metal). As a consequence, a VOF approach is believed to be sufficient;

3. *Incompressible and immiscible fluids*: both the liquid and vapor phases are assumed to be incompressible and immiscible. The first assumption might constitute an unacceptable simplification when dealing with a box pool-type

divertor whose working fluid is lithium at $T_{sat} \lesssim 800\ °C$; in fact, at the nozzle between EC and DC the Mach number $Ma = u/u_s$, that is the ratio of the fluid velocity the sound speed, could be of the order of one, resulting in chocked flow condition, as it is well stated in [4] and [3]. Despite this fact, it should be remembered that the final goal is developing a reliable thermal phase change model; therefore, the assumption of incompressibility is beneficial, because it allows to focus on these phenomena. Once the model is tested on many case studies dealing with boiling/condensation, it can be extended to the compressible case; this should not be difficult in OpenFOAM;

4. *Energy equation sources/sinks*: the effects of gravity are neglected when dealing with the energy equation, as well as the pressure work and dissipation due to viscous stresses. The justification relies on the fact that fluid velocities are expected to be small in a box pool-type divertor. Obviously, test cases which are investigated to verify the code must be consistent with this assumption. Fourier's law is instead used for conduction;

5. *Constant material properties*: for each phase, thermophysical properties are assumed to be constant. Of course, both liquid and vapor thermal conductivities and specific heat capacities depend on temperature. Therefore, polynomial empirical correlations might be implemented. However, it is not known *a priori* how material properties influence the code stability; as a matter of fact, the use of constant quantities eliminates this issue;

6. *Newtonian fluids*: in OpenFOAM, many transport models are available, but in the present thesis all the simulations are performed with Newtonian fluids.

It should also be stressed that, although phase change is under investigation, the energy equation that is implemented is written in terms of temperature and not in terms of enthalpy or interal energy. The main reasons justifying this strong assumption are:

- the model which is used to evaluate the volumetric mass transfer rate is temperature based (see section 2.3);

- from an informatic point of view, the class that is enabled for incompressible solvers only deals with transport (see section 1.1). Thermophysical classes do exist, but are not enabled for the reference solver. This would significantly increase the complexity of the implementation;

2.2 Conservation equations

The set of conservation equations (volume fraction, pressure, momentum and energy) consistent with the aforementioned assumptions is shown below.

$$\begin{cases} \dfrac{\partial \alpha_L}{\partial t} + \nabla \cdot \Big(\vec{u}\, \alpha_L + \alpha_L (1 - \alpha_L)\, \vec{u_r} \Big) - \alpha_L \nabla \cdot \vec{u} = \\ \qquad\qquad = \Big(\dot{m}_{cond}^{'''}\, (1 - \alpha_L) + \dot{m}_{evap}^{'''}\, \alpha_L \Big) \left(\dfrac{1}{\rho_L} - \alpha_L \left(\dfrac{1}{\rho_L} - \dfrac{1}{\rho_G} \right) \right) \\ \\ \dfrac{\partial \rho \vec{u}}{\partial t} - \dfrac{\partial \rho}{\partial t} \vec{u} + \nabla \cdot (\rho \vec{u} \vec{u}) - \nabla \cdot (\rho \vec{u}\,)\, \vec{u} + \nabla \cdot \mathrm{T}_{dev}^{turb} = -\nabla p + \rho \vec{g} + \sigma \kappa \nabla \alpha_L \\ \\ \nabla \cdot \left(\nabla \cdot \Big(\rho \vec{u} \vec{u} \Big) \right) - \nabla \cdot \Big(\nabla p \Big) = \Big(m \dot{m}^{'''}{}_{cond}\, (1 - \alpha_L) + \dot{m}_{evap}^{'''}\, \alpha_L \Big) \left(\dfrac{1}{\rho_L} - \dfrac{1}{\rho_G} \right) \\ \\ \rho\, c_p \left(\dfrac{\partial T}{\partial t} + \nabla \cdot \Big(\vec{u}\, T \Big) - T\, \nabla \cdot \vec{u} \right) - \nabla \cdot \Big(k \nabla T \Big) = \\ \qquad\qquad = H_{fg} \Big(\dot{m}_{cond}^{'''}\, (1 - \alpha_L) + \dot{m}_{evap}^{'''}\, \alpha_L \Big) \end{cases} \tag{2.1}$$

The objective of the following paragraphs is giving a detailed analysis of each equation appearing in the system above.

2.2.1 Volume fraction equation

Since liquid and vapor phases are assumed to be immiscible and not interpenetrating, it appears logical to introduce the liquid volume fraction, that is defined as:

$$\alpha_L(\vec{x}, t) = \frac{V_{Liquid}}{V} = \begin{cases} 1 \text{ if Liquid} \\ 0 \ \leq x \leq 1 \text{ if interface} \\ 0 \text{ if Vapor} \end{cases} \tag{2.2}$$

This quantity is fundamental not only to evaluate the thermophysical properties of the mixture, but also to capture the interface between liquid and vapor, where heat and mass transfer happens. From the numerical point of view, in fact, it can happen that some cells are solved in such a way that is $0 < \alpha_L < 1$. These cells are those defining the moving interface.

The equation that computes this field is the first one of system 2.1 and is here repeated for convenience.

$$\begin{aligned} &\frac{\partial \alpha_L}{\partial t} + \nabla \cdot \Big(\vec{u}\, \alpha_L + \alpha_L (1 - \alpha_L)\, \vec{u_r} \Big) - \alpha_L \nabla \cdot \vec{u} = \\ &\qquad\qquad = \Big(\dot{m}_{cond}^{'''}\, (1 - \alpha_L) + \dot{m}_{evap}^{'''}\, \alpha_L \Big) \left(\frac{1}{\rho_L} - \alpha_L \left(\frac{1}{\rho_L} - \frac{1}{\rho_G} \right) \right) \end{aligned} \tag{2.3}$$

A description of the physical meaning of each term is given below.

- $\dfrac{\partial \alpha_L}{\partial t}$ takes into account the time evolution of the liquid volume fraction;
- $\nabla \cdot \Big(\vec{u}\,\alpha_L\Big)$ and $\alpha_L \nabla \cdot \vec{u}$ are related to transport due to advection. Their difference is equal to $\vec{u} \cdot \nabla \alpha_L$, which is the term to model. However, the numerical implementation of this expression in OpenFOAM is not straightforward and this explains why the difference of two terms is used;
- $\nabla \cdot \Bigg(\alpha_L\,(1-\alpha_L) \cdot \vec{u_r}\Bigg)$ affects the solution only at the interface and represents the isotropic compression contribution which the user can turn on. The relative velocity $\vec{u_r}$ is defined as the difference between the liquid and vapor velocities. As a matter of fact, when phase change is considered, the interfacial mass flux determines a jump condition in density, linear momentum, energy, which must be taken into account by the physical model; in particular, it can be shown that mass flux and relative velocity are directly proportional (see section 2.2.2).The product between liquid and vapor volume fractions is included so that this term is different from zero only at the interface;
- $\Big(\dot{m}^{'''}_{cond}\,(1-\alpha_L)+\dot{m}^{'''}_{evap}\,\alpha_L\Big)\left(\dfrac{1}{\rho_L}-\alpha_L\left(\dfrac{1}{\rho_L}-\dfrac{1}{\rho_G}\right)\right)$ represents the source/sink term due to phase change. Since the volume fraction equation is written with respect to the liquid phase, the condensation rate $\dot{m}^{'''}_{cond}$ is seen as a source term, while the evaporation/boiling term $\dot{m}^{'''}_{evap}$ as a sink. It should also be pointed out that each term is multiplied by its correspondent volume fraction. As a matter of fact, during condensation the physical driver is the amount of vapor that changes phase (during evaporation/boiling it is the other way round) and this explains the presence of $(1-\alpha_L)$ and α_L.

2.2.2 Momentum equation

In a one fluid approach, the momentum equation is solved with respect to the average velocity $\vec{u}$, which is equal to:

$$\vec{u} = \alpha_L\,\vec{u_L} + (1-\alpha_L)\,\vec{u_V} \tag{2.4}$$

where $\vec{u_L}$ and $\vec{u_V}$ are the liquid and vapor velocities.

The physical meaning of the average velocity is clear when α_L is zero or one. Conversely, at the interface $\vec{u}$ is associated to a fictitious third phase that is neither liquid nor vapor. As a consequence, attention should be paid there. When dealing with immiscible and incompressible fluids (these assumptions refer to the single phases and not to the mixture), it can be proved that, whenever phase change is not present, it is:

$$\nabla \cdot \vec{u_L} = \nabla \cdot \vec{u_V} = \nabla \cdot \vec{u} = 0 \tag{2.5}$$

Liquid and vapor velocities are divergence free as a consequence of their mass conservation equations, where incompressibility is assumed. On the other hand, the mixture has also the same property, but this is due to the hypothesis that phase change phenomena are not present. Because of this, the mixture velocity is continuous in all the domain.

When phase change is taken into account, the situation is totally different; in fact, is can be proved that:

$$\nabla \cdot \vec{u} = \dot{m}''' \left(\frac{1}{\rho_L} - \frac{1}{\rho_G} \right) \tag{2.6}$$

where $\dot{m}'''$ is the volumetric mass flow rate, expressed in [kg/(m^3 s)].

Equation 2.6 is the most general expression for the divergence of the mixture velocity (when $\dot{m}''' = 0$ it clearly leads to equation 2.4). From the mathematical point of view, this relation has dramatic consequences at the interface, because it is in this part of the domain that the phase change rate is different from zero. Applying Gauss theorem one can in fact prove that the flux of $\vec{u}$ is not null in this fictitious third phase and this justifies the presence of the term $\nabla \cdot \left(\alpha_L \left(1 - \alpha_L\right) \cdot \vec{u_r} \right)$ in the volume fraction equation, which has been previously discussed.

The momentum equation, that is the second one in the system 2.1 is repeated here.

$$\frac{\partial \rho \vec{u}}{\partial t} - \frac{\partial \rho}{\partial t} \vec{u} + \nabla \cdot (\rho \vec{u} \vec{u}) - \nabla \cdot (\rho \vec{u}\,)\, \vec{u} + \nabla \cdot \mathrm{T}_{dev}^{turb} = -\nabla p + \rho \vec{g} + \sigma \kappa \nabla \alpha_L \tag{2.7}$$

A discussion of the single terms follows.

- $\dfrac{\partial \rho \vec{u}}{\partial t} - \dfrac{\partial \rho}{\partial t} \vec{u}$ is equivalent to $\rho \, \dfrac{\partial \vec{u}}{\partial t}$ and represents the evolution in time of the average velocity. The choice of using the difference of two expressions instead of only one due to numerical reasons;
- $\nabla \cdot (\rho \vec{u} \vec{u}) - \nabla \cdot (\rho \vec{u}\,)\, \vec{u}$ quantifies the advective effect;
- $\nabla \cdot \mathrm{T}_{dev}^{turb}$ is the divergence of the deviatoric stress tensor associated to turbulence effects. These stresses result from the viscosities of the two phases, that are assumed to be constant;
- $-\nabla p$ is the pressure gradient;
- $\rho \vec{g}$ is the volumetric force resulting from gravity;

- $\sigma\kappa\nabla\alpha_L$ represents the effect of the surface tension σ, which applies at the interface between liquid and vapor. The nature of the resulting force is intrinsically superficial, while the momentum equation is written per unit volume; because of this, the multiplicative factor $\nabla\alpha_L$ has to be introduced to take into account the ratio between the interfacial area and the volume. A further parameter, κ, is also required to consider that surface tension effects are significantly relevant wherever the curvature of the interface is high.

2.2.3 Pressure equation

Pressure is not the result of a weighted average, as in the case of $\vec{u}$, but it is continuous all over the domain; in particular, at the interface it is assumed to be equal to the saturation pressure at the local temperature. The pressure equation comes from the combination of the momentum and mass conservation equations (a detailed derivation can be found in [5]). It is the third one of system 2.1 and is here repeated for convenience.

$$\nabla \cdot \left(\nabla \cdot \Big(\rho\vec{u}\vec{u}\Big)\right) - \nabla \cdot \Big(\nabla p\Big) = \Big(\dot{m}_{cond}^{'''}\,(1-\alpha_L) + \dot{m}_{evap}^{'''}\,\alpha_L\Big)\left(\frac{1}{\rho_L} - \frac{1}{\rho_G}\right) \tag{2.8}$$

The first thing to notice is that pressure does neither have time dependencies nor advective terms; only the laplacian operator appears. This is a positive feature because the solution is smoothed; in fact, while divergence operator might lead to discontinuities (e.g. Riemann problem), which are not admissible for this thermodynamic variable, the laplacian smooths the computed field (e.g. thermal diffusion).

The analysis of each term appearing in equation 2.8 is performed below.

- $\nabla \cdot \left(\nabla \cdot \Big(\rho\vec{u}\vec{u}\Big)\right)$ is the advective flux of momentum. Obviously, this term is strongly correlated to the pressure gradient, which represents its driver;
- $-\nabla \cdot \Big(\nabla p\Big)$ is the laplacian of the pressure;
- $\Big(\dot{m}_{cond}^{'''}\,(1-\alpha_L) + \dot{m}_{evap}^{'''}\,\alpha_L\Big)\left(\frac{1}{\rho_L} - \frac{1}{\rho_G}\right)$ quantifies the effect of phase change and is equal to the divergence of the one fluid velocity (see equation 2.6). It seems relevant to point it out that this term does not modify the elliptic nature of the pressure equation; indeed, under the assumption that neither $\dot{m}_{cond}^{'''}$ nor $\dot{m}_{evap}^{'''}$ depend on pressure, 2.8 still has the structure of a Poisson's equation. This is certainly a positive feature, which goes in the direction of smoothing any possible discontinuity that could arise at the interface.

2.2.4 Temperature equation

Since a fundamental aspect of the problem under study is that of phase change, the energy equation should be written either in terms of internal energy or enthalpy. However, as it has already been stated in section 2.1, it is preferable to implement a non conservative form that is temperature based. According to the opinion of the author, this is a very strong assumption; as a matter of fact, specific heat capacities have to be introduced and this might be dangerous when dealing with condensation and/or boiling/evaporation. Despite this, in literature there are many references where the same approach is adopted, such as [6], [7] and [8]; moreover, the thermal phase change model that is implemented requires temperature as an input. Finally, as it has been anticipated, there are some informatic issues related to the use of the transport or thermophysical classes.

As in the case of pressure, at the interface temperature is assumed to be that of saturation; therefore, it is continuous throughout the domain. The equation under analysis is the fourth of system 2.1 and is here repeated.

$$\begin{aligned}\rho\, c_p \left(\frac{\partial T}{\partial t} + \nabla \cdot \Big(\vec{u}\, T \Big) - T\, \nabla \cdot \vec{u} \right) - \nabla \cdot \Big(k \nabla T \Big) = \\ = H_{fg} \left(\dot{m}^{'''}_{cond}\, (1 - \alpha_L) + \dot{m}^{'''}_{evap}\, \alpha_L \right)\end{aligned} \tag{2.9}$$

The first thing to notice is that, consistently with the assumptions of section 2.1, both dissipative effects due to viscous stresses and the gravitational work are absent. The latter is neglected because its contribution is small when compared to that associated to phase change. Pressure work is also neglected for the same reason. Then, it should also be pointed out that the product ρc_p refers to the one fluid and is assumed to be constant (i.e temperature independent). Its value can be easily computed performing a weighted average with respect to the volume fractions; it is in fact:

$$\rho c_p = \rho_L c_{p,L}\, \alpha_L + \rho_V c_{p,V}\, (1 - \alpha_L) \tag{2.10}$$

The description of each term of equation 2.9 follows.

- $\dfrac{\partial T}{\partial t}$ accounts for time evolution;
- $\nabla \cdot \Big(\vec{u}\, T \Big) - T\, \nabla \cdot \vec{u}$ is the advective term and is equal to $\vec{u} \cdot \nabla T$. The choice of writing two terms is related to the implementation in OpenFOAM, exactly as in the case of the correspondent terms in the volume fraction and momentum equations;
- $\nabla \cdot \Big(k \nabla T \Big)$ quantifies the thermal conduction, that is modeled through Fourier's law;

- $H_{fg}\left(\dot{m}^{'''}_{cond}\,(1-\alpha_L)+\dot{m}^{'''}_{evap}\,\alpha_L\right)$ is the source/sink term resulting from phase change. H_{fg} is the vaporization latent heat and in principle should be replaced by the term $\left(H_{fg}+(c_{p,L}-c_{p,V})\,T_{sat}\right)$, as it is explained in [9]; however, the second addend is usually much smaller than the first one and so can be neglected.

2.3 Thermal phase change model

Hardt and Wondra have developed a model to compute the evaporation rate for liquid-vapor flows; the procedure, that is explained in detail in [10] and [6], can be divided in four steps:

1. evaluation of the real mass flow rate $\dot{m}^{'''}_0$. This quantity is different from zero only at the interface, where its value might be very high. Therefore, variations can be very sharp there;

2. computation of a smoothed mass rate: it is the unknown of an Helmholtz's problem with a source that is the mass flow rate calculated at the previous point. The diffusion area multiplying the laplacian is set up by the user; in this way the amount of artificial diffusion that is introduced can be controlled (the smaller the coefficient, the more real - and sharp - the smoothed mass rate);

3. at the interface, the resulting source/sink is put to zero;

4. through a weighting procedure the final mass rate $\dot{m}^{'''}$ is obtained.

As already anticipated in section 1.1, Kunkelmann tried to implement this procedure in OpenFOAM, but some important issues were present (such as non conservation of mass). To the opinion of the author, this might be due to the artificial manipulations done at points 3 and 4 (indeed, these steps are presented as mass conservative). Nevertheless, the way the real mass rate is computed seems to be very physical; furthermore, the key thermodynamic variable that is required to evaluate $\dot{m}^{'''}_0$ is temperature and this would justify the choice of using a thermal energy equation, that is also easier to be implemented than the entalpy/internal energy equation. As a result, it has been decided to evaluate $\dot{m}^{'''}$ using the procedure which computes $\dot{m}^{'''}_0$. Dealing with a sharp source/sink term is the main drawback of this choice; thus, a smoothed interface might be required, in order to have feasible thermophysical variations there.

The approach that is used to compute the thermal phase change mass rate, expressed in [kg/(m^3 s)], is obtained combining Clausius-Clapeyron and Hertz-Knudsen equations; the difference between the local pressure and the saturation one is evaluated through the first relation and it is inserted in the second formula. The final result is:

$$\dot{m}''' = -|\nabla \alpha_L| \frac{2\chi}{2-\chi} \frac{H_{fg}}{\dfrac{1}{\rho_V} - \dfrac{1}{\rho_L}} \frac{1}{\sqrt{2\pi R^*}} \frac{T - T_{sat}}{{T_{sat}}^{3/2}} \tag{2.11}$$

where:

- χ is the fraction of molecules departing from the interface and is in the range (0.04-1.0) (see [11]). In the present thesis its value is one in all the simulations;
- T_{sat} is the saturation temperature;
- R^* is the specific gas constant;
- $|\nabla \alpha_L|$ physically represents the ratio between the interface area and the volume, as it is proved in [10].

The phase change density rate $\dot{m}'''$ is assumed to be positive for condensation, because equation 2.3 refers to the liquid volume fraction (obviously, evaporation is seen as a sink). Since during condensation $T < T_{sat}$, to maintain this convention a "$-$" sign is added to equation 2.11.

2.4 Conclusions

To the knowledge of the author, in OpenFOAM there are still no models that deal with thermal phase change satisfactorily; because of this, a suitable model for that must be implemented by hand.

A one fluid VOF approach is chosen, since dispersion and/or interpenetrating phenomena are not relevant for the purposes of this work; moreover, in literature there are many attempts performed using this kind of techniques. Fluids are assumed to be incompressible, at least for the purpose of testing the methods, and a temperature equation is implemented instead of the internal energy/enthalpy one.

The governing equations, summarized in system 2.1, have been analyzed one by one and the main features of each of them are put in evidence.

Finally, the volumetric mass transfer rate associated to thermal phase change is evaluated. The model by Hardt and Wondra model, which is temperature based, is used to perform this task. No smoothing procedures are performed and this may affect the code stability when solving the equations at the interfacial cells.

References

[1] N. Samkhaniani et al. «A VOF method to phase change modeling». In: *1st iranian conference on heat and mass transfer* (2012).

[2] N. Samkhaniani et al. «The evaluation of the diffusive interface method for phase change simulations using OpenFOAM». 2017.

[3] G.F. Nallo et al. «Modeling the lithium loop in a liquid metal pool-type divertor». In: *Fusion Engineering and Design* (2017).

[4] R.J. Goldston et al. «Recent advances towards a lithium vapor box divertor». In: *Nuclear Materials and Energy* (2017).

[5] M. Peric J.H. Ferziger. «Computation methods for fluid dynamics». 2002.

[6] C. Kunkelmann. «Numerical modeling and investigation of boiling phenomena». 2016.

[7] G. Guedon. «Two-phase heat and mass transfer modeling: flexible numerical methods for energy engineering analyses». 2013.

[8] L. Malan. «Direct numerical simulations of free surface and interfacial flow using the VOF method: cavitating bubble clouds and phase change». 2016.

[9] N. Samkhaniani et al. «Numerical simulation of superheated vapor bubble rising in stagnant liquid». 2017.

[10] F. Wondra S. Hardt. «Evaporation model for interfacial flows based on a continuum field representation of the source terms». In: *Journal of computational physics* (2008).

[11] M. Barheini et al. «Development of a phase change model for the Volume of Fluid Method in OpenFOAM». 2016.

Chapter 3

Implementation of a suitable phase change model in OpenFOAM

OpenFOAM, standing for Open Field Operation And Manipulation, is an open source software written in C++, available for Unix/Linux operating object-oriented systems and released under the GNU General Public License. There are three main different distributors online: OpenCFD, ESIgroup and OpenFOAM Foundation, that was the owner of copyrights at the time of writing; because of this, its latest version OpenFOAM 5.0 is used.

The software consists of many different libraries, which can be divided in two groups. On the one hand, there are all the solvers designed to solve a specific problem of continuum mechanics (i.e. compressible/incompressible single-phase/multiphase flows, stress analysis, heat transfer etc.). On the other hand, there are the utilities required to perform data manipulation (pre-processing, equations numerical resolution, post-processing). A schematic representation of OpenFOAM structure is presented below.

As it can be seen from Figure 3.1, during the solving step one can use either standard applications (their source codes are given by default) or personal applications (compiled by the user according to its aims); another relevant feature is ParaView, the post-processing tool which allows to visualize the results of a simulation.

Each solver is an executable whose source code is a .C file which includes a series of headers .H files. In this way, the code is readable and relatively short (often approximately 100 lines); moreover, each equation has its own script and this let the debug be performed easily.

Particular attention should be given to the way the equation of state is modeled.

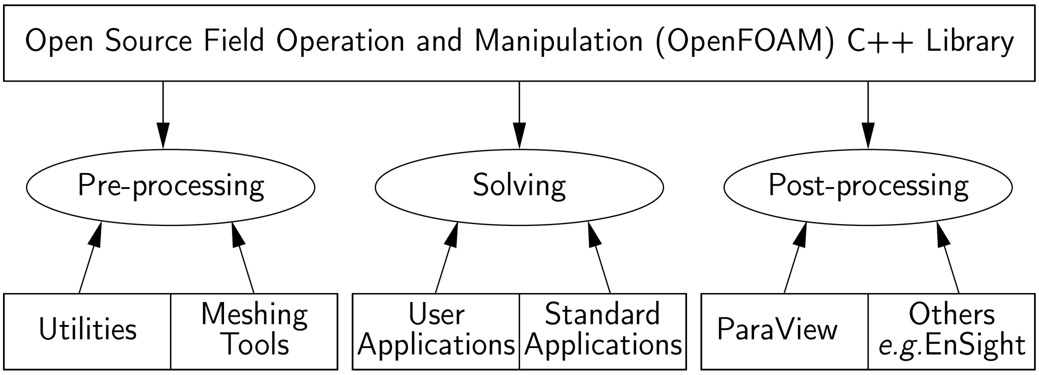

Figure 3.1: Schematic overview of OpenFOAM structure - adapted from [1]

If a solver is designed for incompressible fluids, the only physics that needs to be taken into account is transport; as a consequence, only "transport classes", as they are called in OF nomenclature, are included inside the .C. Conversely, if compressibility is considered, "thermophysical classes" must be used; in fact, temperature becomes important in the definition of the thermodynamic state of the fluid, so transport is no more the only relevant part of the physics. From an informatic point of view, these classes represent the most complex part of the code; in fact, their hierarchical structure is such that different sub-classes can be nested one inside the other, dramatically complicating the global structure of the executable.

3.1 General features of the algorithm

The numerical code implemented to model the governing equations discussed in the previous chapter is based on *interPhaseChangeFoam*, which is a solver available in the package released by OpenFOAM 5.0. interPhaseChangeFoam derives from another executable for multiphase flows, *interFoam*, that is well established and well known to the FOAM community. The main assumptions of these applications are on the type of simulated fluids (incompressible and immiscible) and on the VOF approach; however, while interFoam is just focused on transport phenomena (fluid dynamics), interPhaseChangeFoam also evaluates the mass transfer. Nevertheless, all the implemented phase change models (*Kunz*, *Merkle* and *SchnerrSauer*) only accounts for cavitating phenomena (details can be found in [2]), as opposed to temperature driven phase change phenomena. This explains why the energy equation must be included in order to simulate evaporation/boiling and condensation, thereby extending the use of interPhaseChangeFoam.

Another modification that must be done concerns the interfacial properties; in fact, at the interface between liquid and vapor thermophysical properties feel huge variations (e.g. for water at 1 bar the ratio ρ_L/ρ_V is ~ 1700) and this may cause serious stability problems when performing a CFD simulation. Those cells where it is

$0 \leq \alpha_L \leq 1$ can be badly solved and the error can easily propagate all over the domain, starting from the interface. Because of this, smoothing procedures are provided there in order to deal with affordable gradients.

Of course, one might well argue that this problem is typical of multiphase flows in general; however, there is a fundamental feature characterizing phase change situations: the mixture velocity is not divergence free at the interface (see the previous chapter). This means that:

1. the one fluid (i.e. the mixture) is no more incompressible, even if each phase is;
2. the volume fraction equation has two piece of physics: transport due to advection and source/sink due to phase change. Indeed, when phase change is present the interface cannot be just advected, as in the case of standard fluid dynamics codes.

Algorithm 1: myPhaseChangeFoam code

```
create smooth interface properties
define scalar and vectorial fields
while $(t \leq t_{end})$ do
    while $(n \leq n_{pimple})$ do
        evaluate CFL condition, set $\Delta t$
        update mixture properties
        solve $\alpha_L$ equation
        solve $T$ equation
        correct interface
        solve $U$ equation
        while $(i \leq n_{corr})$ do
            solve $p_{rgh}$ equation
        end
        if (turbulence on) then
            correct for turbulence
        end
    end
end
```

A further complication is represented by *spurious currents*, which are typical of surface tension dominated flows. As it is well stated in [3], [4], [5] and [6], they are associated to unphysical velocities near the interface resulting from the numerical errors in calculating the normal vector which is needed to compute the curvature κ. Clearly, spurious velocities enhance the mass flux caused by the thermal phase change, but they have no physical meaning, because they come from the numerics.

The solver used in the present thesis is named *myPhaseChangeFoam*. The code is based on the algorithm shown below, and is analyzed in the following paragraphs.

3.2 Fields definition

createFields.H is the header file where all the main fields used in the code are declared. Since OpenFOAM is based on Finite Volume Method (FVM) and adopts a collocated arrangement, usually fields are defined with respect to the cell centre; for instance, Listing 3.1 shows how pressure and velocity are defined.

```
volScalarField p_rgh                          volVectorField U
(                                             (
    IOobject                                      IOobject
    (                                             (
        "p_rgh",                                      "U",
        runTime.timeName(),                           runTime.timeName(),
        mesh,                                         mesh,
        IOobject::MUST_READ,                          IOobject::MUST_READ,
        IOobject::AUTO_WRITE                          IOobject::AUTO_WRITE
    ),                                            ),
    mesh                                          mesh
);                                            );
```

Listing 3.1: Definition of pressure and velocity fields

It can be noticed that:

- both U and p_{rgh} must be read at $t = 0$ s ("MUST_READ"), are computed at each timestep ("runTime.timeName()") and are automatically written at those time set up by the user ("AUTO_WRITE"). Moreover, both fields refer to the volume ("volScalar(Vector)Field") of each cell of the mesh ("mesh");
- p_{rgh} is the sum between the pressure p and the hydrostatic contribution ρgh. The equation solved in Algorithm 1 is in fact written in terms of p_{rgh}.

An interpolation procedure is required to compute the fluxes through the boundary faces (fluxes are "surfaceScalarFields"). This numerical approximation might affect the splitting error characterizing the PISO algorithm and can be eliminated by using a staggered arrangement (this is not the case of OpenFOAM). It is important to stress that fluxes are in [m^3/s] and not in [kg/s]; to the opinion of the author, this choice is questionable and it is due to the fact that interPhaseChangeFoam derives from interFoam, which assumes incompressible flow for both the single phases and for the mixture. In interPhaseChangeFoam, however, the one fluid is not divergence free, so mass fluxes are not just the volumetric ones times a constant density. However, implementing a different expression would require deep modification to the code and a very deep knowledge of numerics, because divergence operator in OpenFOAM is defined on surfaceScalarField (see section **??**) and not on vectors (as it should be). Because of all these reasons, no modifications have been performed concerning the way fluxes are defined.

With respect to the reference solver, an additional field, temperature, is added. Its definition is equivalent to p_{rgh}; other small variations are due to the declaration of the specific heat capacities and thermal conductivities of the two phases (they are defined as "dimensionedScalar").

3.3 Equations

Volume fraction and pressure, equations are the main responsible for phase change. Unfortunately, they are also by far the most complicated ones. On the one hand, α_L is the only quantity that is bounded from above and from below (it must be $\alpha_L \in [0,1]$) and this introduces a constraint; on the other hand, pressure is fundamental in fluid dynamics problems and may be responsible of many issues, caused by the splitting algorithm through which the coupling $p - U$ is solved.

Despite this, it is certainly not possible to put the temperature variable in the background, since it is the driver of the volumetric mass transfer rate; indeed, the whole problem is strongly coupled and this is reflected by the order according to which the single equations are solved in Algorithm 1. With respect to interPhaseChangeFoam, the temperature equation is added and it has to be collocated at the correct place within the algorithm. One may in principle think that, when using a PIMPLE resolving algorithm (that means $n_{pimple} \geq 2$ in Algorithm 1 - see [1] for details), this equation can be put at any place inside the outer while loop, because for each time step all the set of conservation equations is solved more than once. This is unfortunately not correct. As said, the system of non linear partial differential equations is strongly coupled and the position of each equation can dramatically influence the code stability; indeed, after many trials the order shown in Algorithm 1 has been found to be the best and this is justified by the following considerations:

- the $p - U$ coupling is solved using the temperature field of the particular time step under consideration. It is therefore not a good device to put the T equation between U and p because they are solved almost together (PISO - that means $n_{pimple} = 1$ in Algorithm 1 - and/or PIMPLE algorithms). This explains why T is placed before p and U;

- for stability issues, the first equation to be solved must be that of the volume fraction. Indeed, this quantity has a key role in the evaluation of the mass transfer rate because it is precisely the term $|\nabla \alpha_L|$ that causes the very sharp variation of $\dot{m}'''$ at the interface.

The implementation of each equation is discussed in the following paragraphs.

3.3.1 Volume fraction

The α_L equation is solved in a file called *alphaEqn.H*, which is present inside another header, *alphaEqnSubCycle.H*; the user can in fact decide how many times the volume fraction equation must be solved at each outer loop of the PISO/PIMPLE algorithm. Its numerical time step is evaluated as $\Delta t_\alpha = \Delta t / N_{sub}$, where N_{sub} is the number of subcycles and Δt the actual time step (see [4]).

When solving for the volume fraction, the Multigrid Universal Limiter for Explicit Solution (MULES) is applied; this is an algebraic bounding procedure which limits the alpha fluxes and is fundamental for the approach followed within this thesis (a detailed analysis can be found in [7]). VOF methods can in fact be divided in two groups: algebraic, such as MULES, and geometric, such as Piecewise Linear Interface Calculation (PLIC - see [4] and [6]). The choice of the method used to advect α_L has a dramatic impact on the code, because it can strongly influence both its stability and accuracy. Because of this, Malan proposed in [8] a new approach, that relies on the idea of solving first a pure advection equation using a standard geometric VOF method; then, a second step is required to take into account the shift of the interface due to phase change. Although the procedure is potentially interesting, it is still under development, as Zaleski stated in [9].

OpenFOAM only supports MULES and this is, according to the opinion of the author, a huge limitation for a CFD software which intends to deal with multiphase flows. Indeed, as it is stated in the description given in "MULES.H", this algebraic procedure is for a convective-only transport equation and this also explains why many attempts have been made to implement other VOF methods. However, none of them has been found to be sufficiently reliable, as cit will be made clear in the following chapter.

```
fvScalarMatrix alpha1Eqn
(
     fv::EulerDdtScheme<scalar>(mesh).fvmDdt(alpha1)
   + fv::gaussConvectionScheme<scalar>
     (
         mesh,
         phi,
         upwind<scalar>(mesh, phi)
     ).fvmDiv(phi, alpha1)
   - fvm::Sp(divU, alpha1)
   ==
     fvm::Sp(vDotvmcAlphal, alpha1)
   + vDotcAlphal
);
```

Listing 3.2: Numerical implementation of the volume fraction equation

The volume fraction equation can be solved in two different ways, depending on the value of the boolean "MULESCorr". If switched on, a correction is applied after the numerical resolution; otherwise, the "explicitSolve" template, available in "MULESTemplate.C" is used. Listing 3.2 shows how the α_L equation is implemented when MULESCorr is set to true.

Some comments can be done:

- "phi" [m^3/s] evaluates the flux of velocity through the boundary faces of each cell and contains the interfacial compressive effect given by the relative velocity $\vec{u_r}$;
- "divU" stands for "divergence of U" but can not model according to the expression discussed in the previous chapter; in fact, that part of the physics, which characterize phase change, is taken into account inside the variables "vDotvmcAlpha1" and "vDotcAlpha1". Therefore, "divU" is just the divergence of "phi". This unhappy nomenclature comes from the fact that the reference solver derives from another one, interFoam, where the one-fluid is divergence free and therefore there is no risk of misunderstandings;
- "vDotcAlpha1" [1/s] models the condensation rate, while "vDotvmcAlpha1" is defined as the difference between "vDotcAlpha1" and the evaporating rate "vDotvAlpha1";
- "fvm::Sp" stands in this case for "implicit product". This is the way OpenFOAM uses to multiply a known scalar field with the unknown quantity which must be evaluated.

3.3.2 Temperature

The T equation is solved in the file *TEqn.H*, which is not part of the reference solver and is therefore implemented from scratch. Its code is reported below.

```
fvScalarMatrix TEqn
(
    fvm::ddt(T)
  + fvm::div(phi, T)
  - fvm::Sp(fvc::div(phi),T)
  - fvm::laplacian( k/rhoC , T,"laplacian(alphaEff,T)")
 ==
    fvm::Sp((vDotcT-vDotvT)/rhoC,T) - (vDotcT-vDotvT)*TSatLocal/...
   rhoC
);
```

Listing 3.3: Numerical implementation of the temperature equation

Let’s analyze what is in Listing 3.3:

- the thermal conductivity "k" [W/(m K)] and the heat capacity "rhoC" [J/(m^3 K)] are those of the one fluid and are evaluated by performing a weighted average through the volume fraction, which has been computed immediately before (see Algorithm 1). Since $\rho_L >> \rho_V$ and $c_{p,L} \sim c_{p,V}$, it is clear that "rho1*C1" is much bigger than "rho2*C2"; using the updated field of α_L is thus very important in order to properly solve the temperature equation, thereby avoiding possible unwanted instabilities which may arise. In equation 3.3, all the terms are divided by "rhoC", since it is assumed to be constant;

- "fvm::ddt" models the time Eulerian time derivative ("fvm::DDt" is instead the function used for material derivatives);

- the divergence operator is evaluated by the function "fvm::div", which applies to a "surfaceScalarField". As a consequence, the T field is interpolated on the boundary faces and the result is multiplied by "phi", which is already defined on the cell surfaces;

- the laplacian term quantifies the heat conduction and is composed of two parts: "k/(rhoC)" is always present, while "alphaEff" is the thermal diffusivity due to turbulence. Obviously, its value is zero when laminar simulations are performed;

- the right hand side quantifies the mass transfer due to condensation ("vDotcT") and evaporation ("vDotvT"). These quantities do not incorporate the difference $(T - T_{sat})$, present in Hardt and Wondra model, and this explains why in equation 3.3 there are two addends: one multiplies the variable "T", the other one "TSatLocal", which is constant. Through this implementation, the temperature value appearing in the phase change model is implicit, so its coefficient is added to the diagonal of the matrix obtained after the numerical discretization is done.

3.3.3 Pressure - velocity coupling

Segregated solvers compute the general Navier-Stokes equation iteratively. In particular, most of the applications in OpenFOAM use the Pressure Implicit Split Operator (PISO), the Semi Implicit Method for Pressure Equations (SIMPLE - it is used for steady state problems), or the combined PIMPLE (see [1]). These algorithms all solve the pressure and velocity equations in a coupled way; the procedure is summarized below.

1. U equation is solved using the pressure distribution computed at the old time step;
2. the computed velocity U^* is used to evaluate the flux of pressure;
3. p equation is solved and the new pressure distribution is obtained;

4. a correction term is added to U^* in order to get the new velocity field U.

Further details can be found in [10]. It should be noticed that, due to the presence of $\dot{m}'''$, the decoupling existing between continuity and momentum equations may result is oscillations arising at the interface. This issue is not present in staggered grids, where the velocity field is exactly computed at the boundary faces (therefore, interpolation errors are eliminated "by design").

```
fvVectorMatrix UEqn
(
    fvm::ddt(rho, U)
  + fvm::div(rhoPhi, U)
  - fvm::Sp(fvc::ddt(rho) + fvc::div(rhoPhi), U)
  + turbulence->divDevRhoReff(rho, U)
);
```

Listing 3.4: Numerical implementation of the momentum equation

Listing 3.4 shows the implementation of the momentum equation, that is present in *UEqn.H*. It can be noticed that the surface tensions, pressure gradient and gravity forces do not appear in the definition of the vector matrix "UEqn"; indeed, they are seen as source/sink terms and are evaluated if and only if the boolean called "momentumPredictor" is switched on (its name is self explanatory), as it is clear by looking at Listing 3.5.

```
if (pimple.momentumPredictor())
{
    solve
    (
        UEqn
      ==
        fvc::reconstruct
        (
            (
                interface.surfaceTensionForce()
              - ghf*fvc::snGrad(rho)
              - fvc::snGrad(p_rgh)
            ) * mesh.magSf()
        )
    );
}
```

Listing 3.5: Effect of momentum predictor on the momentum equation

In principle, one might think that the boolean predictor must always be set up to one; however, analyzing the header file *pEqn.H* it can be concluded that this statement is wrong. Indeed, assuming that the momentum equation can be written in matrix form as:

$$A\,U = H \tag{3.1}$$

where A is a matrix and H a "volVectorField" (exactly as U), it is easy to understand the physical meaning of the variable defined as "HbyA": it is the predictive velocity U^*. Its flux is called "phiHbyA" and is the important quantity entering the pressure equation. Indeed, "phiHbyA" also contains another term, called "phig", which accounts for surface tension and gravity. In brief, if the switch "momentumPredictor" is set to true, further calculations must be performed and these operations may significantly affect the numerical stability of the whole code.

The implementation of the Poisson's equation evaluating the pressure distribution in the computational domain is reported in Listing 3.6.

```
fvScalarMatrix p_rghEqn
(
    fvc::div(phiHbyA)
  - fvm::laplacian(rAUf, p_rgh)
  - (vDotvP - vDotcP)*(mixture->pSat() - rho*gh)
  + fvm::Sp(vDotvP - vDotcP, p_rgh)
);
```

Listing 3.6: Numerical implementation of the pressure equationn

The variables "vDotcP" and "vDotvP" quantify the effect of condensation and evaporation. With respect to the values obtained from the model of Hardt and Wondra, they are explicitly divided by the factor $p_{rgh} - p_{sat} + \rho g h$, which is then multiplied implicitly to define the matrix "p_rghEqn"; this passage is a numerical trick done in the reference solver in order to guarantee code stability and has been maintained in myPhaseChangeFoam.

3.4 Thermophysical properties

In interPhaseChangeFoam, the transport properties of the mixture are computed inside a class named *phaseChangeTwoPhaseMixture*, which in turn also calls the phase change model the user has selected. As already anticipated in section 3.1, in the reference solver there are only three options, all dealing with cavitation. As a consequence, the model by Hardt and Wondra must be implemented from scratch. In this respect, OpenFOAM hierarchical structure turns out to be very useful, because it allows the programmer to add whatever he/she wants at high level without modifying the kernel of the code. Indeed, other thermal phase change models have been implemented (i.e. Lee), but very few tests have been performed using these models since they were found to give more inaccurate results than those of Hardt and Wondra.

In the previous section, the implementation of the governing equations has been discussed and the source/sink terms associated to thermal phase change have been qualitatively analyzed; however, the code written to compute them was not shown and is the topic of the next paragraph. After that, the smooth interface modeling is described.

3.4.1 Phase Change

Listinf 3.7 shows how the source/sink terms for the volume fraction equation are computed. They have not been exactly the same quantities appearing in *alphaEqn.H*, because they are not multiplied by the constant coefficient depending on the densities of the phases yet.

```
Foam::Pair<Foam::tmp<Foam::volScalarField> >
Foam::phaseChangeTwoPhaseMixtures::Hardt::mDotAlphal() const
{
    const volScalarField& T = alpha1_.db().lookupObject<...
    volScalarField>("T");
    return Pair<tmp<volScalarField> >
    (
      -rc_*Cm1_*min(T - TSatLocal() ,T0_)*AbyV()/sqrt(pow(TSatLocal...
    () ,3.0))
      ,
        -rv_*Cm1_*max(T - TSatLocal() ,T0_)*AbyV()/sqrt(pow(...
    TSatLocal() ,3.0))
    );

}
```

Listing 3.7: Source/sink terms for the volume fraction equation

A brief discussion of each term follows:

- "rc_" ("rv_") is a flag whose value is zero when condensation (boiling) is absent. In principle, it is useless because of the way the temperature difference $T - T_{sat}$ is evaluated ("T0_" is $0\ K$). However, the value of the flag is set up by the user and this might be useful in order to be sure which phenomena are certainly neglected; in fact, it may happen that, although a pure evaporating case is under study, some cells read a temperature smaller than the saturation one because of previous calculations. Thanks to "rc_", these potential numerical errors are eliminated by design;

```
Cm1_("Cm1_", 2.0*Cv_*Hfg_*(rho1_*rho2_)/(rho1_-rho2_)/((2.0-Cv_...
    )*pow(2.0*M_PI*R_,0.5)))
Foam ::volScalarField Foam::phaseChangeTwoPhaseMixtures::Hardt...
    ::AbyV() const
{
  volScalarField limitedAlpha1 = min(max(alpha1_, scalar(0)), ...
    scalar(1));
  return( mag(fvc::grad(limitedAlpha1)) );
}
```

Listing 3.8: Definitions of the coefficient Cm1_ and the function AbyV()

- "Cm1_" is the constant coefficient coming from the physical model (see Listing 3.8) and depends on latent heat, elastic coefficient, densities of both the phases (all defined by the user);
- "AbyV()" is the function evaluating $|\nabla\alpha|$ (see Listing 3.8);

```
Foam::Pair<Foam::tmp<Foam::volScalarField> >
Foam::phaseChangeTwoPhaseMixtures::Hardt::mDotP() const
{
    const volScalarField& T = alpha1_.db().lookupObject<...
   volScalarField>("T");
    const volScalarField& p = alpha1_.db().lookupObject<...
   volScalarField>("p");
    volScalarField limitedAlpha1 = min(max(alpha1_, scalar(0)), ...
   scalar(1));
    return Pair<tmp<volScalarField> >
    (
      -rc_*Cm1_*min(T - TSatLocal(),T0_)*AbyV()/sqrt(pow(TSatLocal...
   (),3.0)) * pos(p-pSat_)/max(p-pSat_,1E-6*pSat_) * (1.0-...
   limitedAlpha1)
      ,
      -rv_*Cm1_*max(T - TSatLocal(),T0_)*AbyV()/sqrt(pow(TSatLocal...
   (),3.0)) * neg(p-pSat_)/max(pSat_-p,1E-05*pSat_) * limitedAlpha1
    );
}
```

Listing 3.9: Source/sink terms for the pressure equation

```
Foam::Pair<Foam::tmp<Foam::volScalarField> >
Foam::phaseChangeTwoPhaseMixtures::Hardt::mDotT() const
{
    const volScalarField& T = alpha1_.db().lookupObject<...
   volScalarField>("T");
    volScalarField limitedAlpha1 = min(max(alpha1_, scalar(0)), ...
   scalar(1));
    return Pair<tmp<volScalarField> >
    (
           -rc_*Cm1_*AbyV()/sqrt(pow(TSatLocal(),3.0)) * neg(T - ...
   TSatLocal()) * (1.0-limitedAlpha1)
           ,
           rv_*Cm1_*AbyV()/sqrt(pow(TSatLocal(),3.0)) * ...
   limitedAlpha1 * pos(T - TSatLocal())
    );
}
```

Listing 3.10: Source/sink terms for the temperature equation

When dealing with the source/sink terms of the remaining equations (temperature and pressure) the same comments as before apply; in particular, they are not exactly equal to the corresponding quantities appearing in *TEqn.H* and *pEqn.H*, as they should be multiplied by a constant coefficient (latent heat for temperature, difference of specific volumes for pressure). Listing 3.9 and 3.10 show the code implementation.

3.4.2 Smooth interface

As it is now clear, the resolution of the governing equations is a particular issue at those cells where $0 \leq \alpha_L \leq 1$; therefore, a procedure aimed to smooth the sharp interface must be used. Inside *phaseChangeTwoPhaseMixture* there are many sub-classes, such as *interfaceProperties*, that models this specific part of the computational domain. What has been done in the present solver is to modify "createInterface" such that a smoothing procedure is added; the resulting new sub-class is called *smoothInterfaceProperties*.

As it has already been discussed, the curvature may determine unphysical spurious currents which can affect the whole solution, expecially for tension dominated flows. Therefore, the main modification is carried out on the curvature vector, κ, which is defined as:

$$\kappa = \frac{\nabla \tilde{\alpha_L}}{|\nabla \tilde{\alpha_L}|} \tag{3.2}$$

where $\tilde{\alpha_L}$ is computed according to Lafaurie's smoother, whose implementation is shown in Listing 3.11. "smoothAlpha_", that is $\tilde{\alpha_L}$ of equation 3.2, is computed as the surface weighted average of the volume fraction inside a for loop whose limit, "smoothItr_", is set up by the user.

```
for (int i = 0; i < smoothItr_; ++i)
{
    //Lafaurie smooth function
    smoothAlpha_ = fvc::average(fvc::interpolate(smoothAlpha_));
}
```

Listing 3.11: Lafaurie's smoother for the volume fraction

Before evaluating the curvature, a smooth function is defined and used to evaluate "K_". The variable "fvc::div(nHatf_)" is equal to "K_" of equation 3.2 ("fvc" stands for "explicit product", since "nHatf_" is known). As in the case of $\tilde{\alpha_L}$, the repetition condition ("i < kSmoothItr_" is set up by the user. Listing 3.12 shows the corresponding code.

```
//smoother for curvatuare
```

```
volScalarField smoothFunction = 2.0*sqrt (mag(smoothAlpha_*(1.0 - ...
    smoothAlpha_)))+1E-30;
volScalarField Ks = K_*0.0;
for (int i = 0; i < kSmoothItr_; ++i)
{
  Ks = fvc::surfaceSum(fvc::interpolate(K_*smoothFunction))/fvc::...
    surfaceSum(fvc::interpolate(smoothFunction));
  K_ = -smoothFunction*fvc::div(nHatf_) + (1.0 - smoothFunction)*Ks...
    ;
}
```

Listing 3.12: Evaluation of the curvature

It seems relevant to notice that, whenever "smoothAlpha_" and "ksmoothItr_" are both zero, the standard treatment of the interface is retrieved.

3.5 Conclusions

In this section, the numerical implementation of the code has been discussed. The reference solver released by OpenFOAM only deals with cavitating phenomena; therefore, the temperature equation and the model of condensation and evaporation/boiling must be included.

The strong coupling existing among the equations lead to a specific order, that is summarized in Algorithm 1; in particular, the first equation that must be solved is that of the volume fraction, while the pressure-velocity coupling must be at the end of the while loop.

The equations have been analyzed and the main code lines of each of them have been reported to let the reader better understand what has been done.

Finally, the problem associated to the interface, which is treated by means of the MULES algebraic method, has been discussed in detail and a smooth procedure has been implemented in order to reduce as much as possible the spurious currents that are present there.

References

[1] The OpenFOAM Foundation. *User guide - version 5.0.* 2017.

[2] G. Cazzoli et al. «Assessment of the cavitation models implemented in OpenFOAM under DI-like conditions». In: *71st conference of the Italian machines engineering association* (2016).

[3] F. Wondra S. Hardt. «Evaporation model for interfacial flows based on a continuum field representation of the source terms». In: *Journal of computational physics* (2008).

[4] C. Kunkelmann. «Numerical modeling and investigation of boiling phenomena». 2016.

[5] N. Samkhaniani et al. «The evaluation of the diffusive interface method for phase change simulations using OpenFOAM». 2017.

[6] G. Guedon. «Two-phase heat and mass transfer modeling: flexible numerical methods for energy engineering analyses». 2013.

[7] S.M. Damian. «An extended mixture model for the simultaneous treatment of short and long scale interfaces». 2013.

[8] L. Malan. «Direct numerical simulations of free surface and interfacial flow using the VOF method: cavitating bubble clouds and phase change». 2016.

[9] S. Zaleski. «Volume Of Fluid and lattice-Boltzmann methods». 2018.

[10] M. Peric J.H. Ferziger. «Computation methods for fluid dynamics». 2002.

Chapter 4

Verification

In order to verify the numerical model discussed in the previous chapter, relevant case studies are here analyzed. The evaporating and condensating 1D Stefan problems are firstly solved; for both of them, analytical expressions evaluating the position of the interface as a function of time are available. The classical Nusselt vertical plate problem is then reproduced and the liquid film width at the outlet is compared with that resulting from the analytical formula. At this point, mass conservation is put under investigation and important issues are discovered. This confirms that VOF methods might fail when dealing with phase change.

4.1 1D evaporating Stefan problem

The one dimensional evaporating Stefan problem is a typical benchmark that is used in the verification procedure of numerical codes which deal with boiling/evaporation. At the beginning of the simulation, the domain is completely filled by the liquid phase, which is at saturation conditions. A wall that is at a temperature $T_w > T_{sat}$ transfers heat by conduction to the stagnant fluid and this causes boiling. In a VOF approach, since the different phases are immiscible and interpenetrating, the liquid is pushed out from the domain as soon as vapor is formed. Density dramatically decreases at the cells where phase change occurs and this results in a local expansion produced by the moving interface. Therefore, outflow is expected, even if it is not due to advection due to the particular nature of the problem.

Thanks to its simplicity, both in the physics and in the geometry, the 1D Stefan problem is a good starting point to perform the verification procedure of the developed code.

4.1.1 Simulation setup

Figure 4.1 shows the mesh and the boundary conditions (BCs) of the simulation.

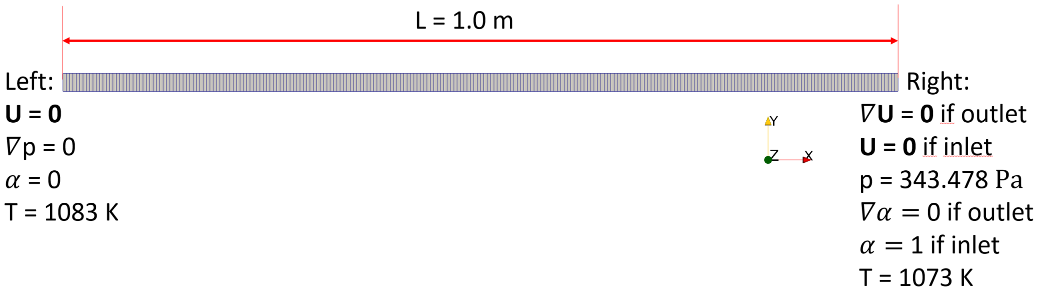

Figure 4.1: Mesh and boundary conditions for 1D Stefan Problem

Since OpenFOAM only admits 3D objects, to get a 1D geometry the mesh is performed in such a way that one cell is present in both the y-axis and z-axis. Thanks to this trick, it is possible to set up an *empty* type BCs to the useless surfaces, so that only *left* and *right* need to be specified. In the present simulation, the x-axis is divided in 250 parts, so the domain is composed of 250 cells, which are all equal, since the expansion factor is assumed to be one.

The working fluid is lithium, which is initially all liquid at uniform $p = p_{sat}$ and $T = T_{sat}$. Table 4.1 shows the thermophysical properties of both the liquid and vapor phases associated to this thermodynamic state.

Table 4.1: Thermophysical properties of liquid and vapor lithium

Quantity	Symbol	Value	Unit
saturation pressure	p_{sat}	343.478	Pa
saturation temperature	T_{sat}	1073	K
vaporization latent heat	H_{fg}	$19590.5e3$	J/kg
elastic gas coefficient	R	1197.23	$J/(kg\ K)$
surface tension	σ	0.3013	N/m
liquid kinematic viscosity	ν_L	$5.08811e-7$	m^2/s
liquid density	ρ_L	454	kg/m^3
liquid specific heat capacity	$c_{p,L}$	4160	$J/(kg\ K)$
liquid thermal conductivity	k_L	56.0	$W/(m\ K)$
vapor kinematic viscosity	ν_V	0.037097	m^2/s
vapor density	ρ_V	$0.278409e-3$	kg/m^3
vapor specific heat capacity	$c_{p,V}$	1797	$J/(kg\ K)$
vapor thermal conductivity	k_V	0.115941	$W/(m\ K)$

By looking at the numerical values reported in the Table above, it is immediately clear how peculiar the fluid properties are. Let us focus, for example, on the vapor density: assuming the ideal gas law is applicable (this is actually not true, but it is useful to perform a first rough estimate), it is immediately clear that, due to the high vacuum conditions (pressure is very low), density is of the order of g/m^3. The ratio ρ_L/ρ_V (and, as a consequence, ν_L/ν_V) is therefore of the order of 10^6 (10^5) and this is a very bad news; when solving for those cells where $0 < \alpha_L < 1$, the huge difference in thermophysical properties may enhance the numerical issues discussed in the previous chapter and this could affect both the stability and the accuracy of the code, that is fluid dependent.

Surface tension is about four times higher than that of water at standard conditions and this may affect the resolution of the $p-U$ coupling in those cells where the curvature is high. The smoothed procedure implemented for the interface may be required; however, it is not used for this simulation, because the interface is plane.

The working fluid is a metal, so thermal conductivities of both the phases are high.

The left surface is the hot wall; its temperature is assumed to be $10\ K$ over saturation, while the liquid volume fraction is assumed to be zero. Imposing such a BC guarantees that at $t = 0\ s$ phase change is different from zero, letting boiling start. This occurs because the volumetric mass transfer rate is in fact proportional to $|\nabla\alpha_L|$. A possible alternative would be setting an homogenous Neumann BC for α_L, thereby initializing a film of vapor.

Velocity is zero on the left surface, while on the right an homogenous Neumann condition is set up, unless reverse flow happens; in this case, a zero velocity is imposed. In principle, this physical situation should never occur; however, when solving the equations it might happen that an ingoing flux is computed on the outlet. If this happens, imposing a null velocity would prevent any rising issues. In other words, the *inletOulet* conditions introduces less constraints than a simple homogeneous Neumann.

Pressure has a zero gradient on the left, while it has a fixed value on the right ($p = p_{sat}$). Since the solver assumes incompressibility, $p = 0\ Pa$ would be the best option; however, according to the implementation of the p equation (see previous chapter), the p equation is solved for the relative pressure and this explains why such a choice is done.

4.1.2 Results

The analytical formula expressing the interface position δ as a function of time t is derived in [1] and here reported for convenience:

$$\delta(t) = 2\eta\sqrt{\alpha_{D,V}t} \tag{4.1}$$

where $\alpha_{D,V}$ is the vapor thermal diffusivity, while η is computed as:

$$\eta\, e^{\eta^2}\, erf(\eta) = \frac{c_{p,V}\,(T_w - T_{sat})}{\sqrt{\pi}H_{fg}} \tag{4.2}$$

T_w being the wall temperature. In the present case, it is $\eta = 0.02141$.

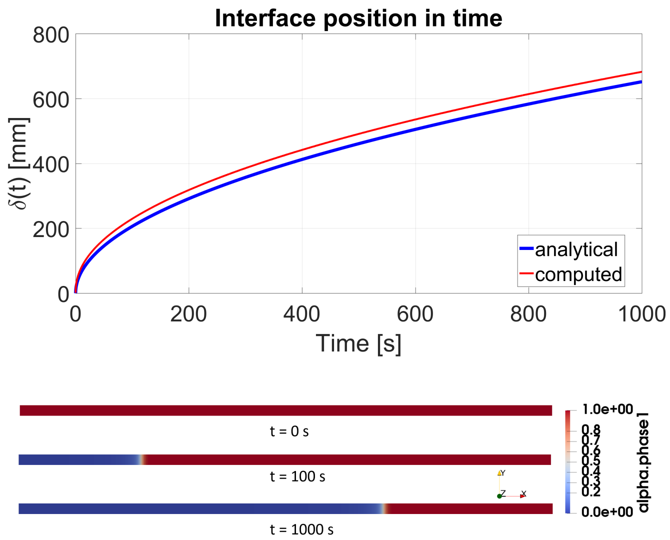

Figure 4.2: On the top, the position of the interface as a function of time. On the bottom, the liquid volume fraction inside the domain at three different times.

Figure 4.2 shows the evolution in time of the interface position between liquid and vapor. As it can be seen, a good agreement exists between the analytical formula and the result of the computation. Clearly, the speed at which δ moves is higher at small times, because the heat transfer from the wall quickly reaches the front of boiling; as time passes, the amount of vapor increases, so heat conduction has to travel a longer distance. It can also be noticed that the interface is not clearly defined, as it should be since VOF is used; however, this feature is only due to the lack of accuracy resulting from the coarse mesh. A better refinement would lead to a much sharper interface, as it would be expected.

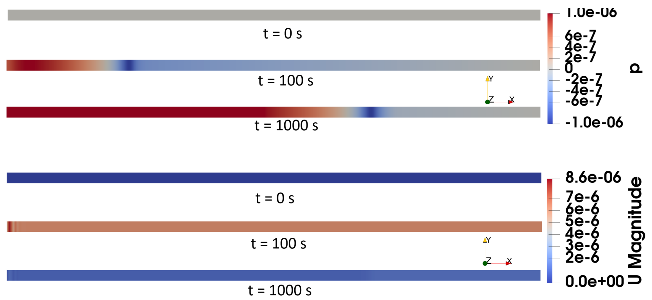

Figure 4.3: On the top, the pressure distribution, computed as $p - p_{sat}$ is shown for three different times. On the bottom, the velocity magnitude associated to the same times is reported.

Figure 4.3 shows the pressure and velocity distributions inside the domain at three different times. At the interface there is a pressure sink which is due to the significant density reduction associated to the mass transfer. This results in a fluid motion, which is however really small, as it is clear by looking at the numerical values of the velocity magnitude. To the opinion of the author, the fluid motion only has a numerical nature and is negligible.

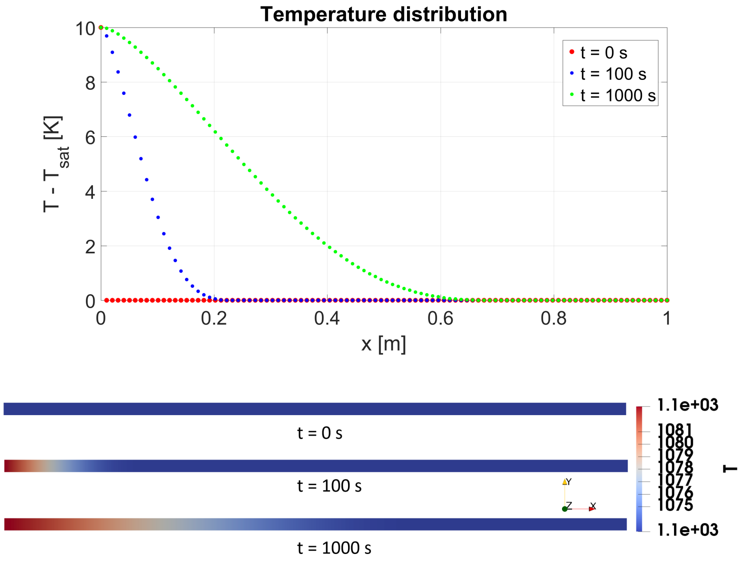

Figure 4.4: Temperature distribution inside the domain at three different times

The temperature distribution inside the domain is shown in Figure 4.4; in particular, by looking at the top plot the typical shape of a diffusive problem can be noticed: at the initial time the distribution is a Dirac delta, while for times $t > 0\ s$ it is a gaussian function whose standard deviation increases in time.

At this point, it seems relevant to notice that the simulation is performed using an adaptive time step, which is limited by either the Courant-Friedrichs-Lewy (CFL) condition (a maximum value of 0.8 is assumed), or an upper bound ($\Delta t_{max} = 0.1\ s$). Because of this, a time convergence study is believed to be not required and this explains why only the grid convergence study is done. Figure 4.5 summarizes the results, explaining why the case of 250 cells has been chosen. The relative error is computed as:

$$err\% = \frac{|\delta_{analytical} - \delta_{computed}|}{|\delta_{analytical}|} \tag{4.3}$$

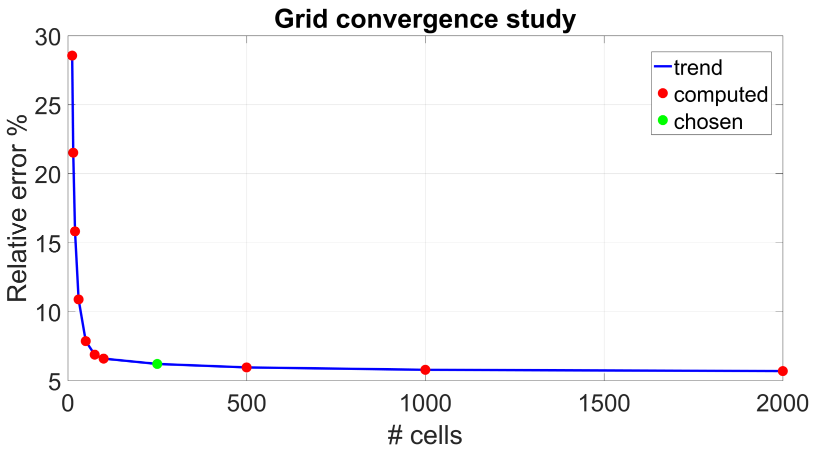

Figure 4.5: Grid convergence study

4.2 1D condensating Stefan problem

The one dimensional condensating Stefan problem is the opposite case with respect to the previous benchmark: at the beginning of the simulation the domain is all filled by saturated vapor. A cold wall at a temperature $T_w < T_{sat}$ causes condensation, and therefore the one fluid density significantly increases in those cells feeling the phase change. The resulting local contraction due to the moving interface is such that the velocity field goes in the opposite direction (i.e. it is negative).

4.2.1 Simulation setup

Mesh geometry and BCs are in Figure 4.6 (the fluid is the same as before). They are equal to those of evaporation, with the following exceptions:

- T_w is $30\ K$ lower than T_{sat};
- α_L is one on the left surfaces, zero on the right (only effective in case of back flow);
- at the initial time $t = 0\ s$ the domain is all filled by vapor at uniform $p = p_{sat}$ and $T = T_{sat}$.

Further comments that can be done are associated to the amount of cells (1000) and to the length of the domain (see next paragraph). Although condensation is, from the physical point of view, the opposite of evaporation, to perform a CFD study is much more complicated than before. The left hand side of the volume fraction equation is in fact null at $t = 0\ s$, so the volumetric mass transfer rate definitely represents the only relevant piece of physics, as well as the driver of the simulation. As a matter of fact a 1D condensating Stefan problem is computationally much more expensive than an evaporating one which has the same mesh.

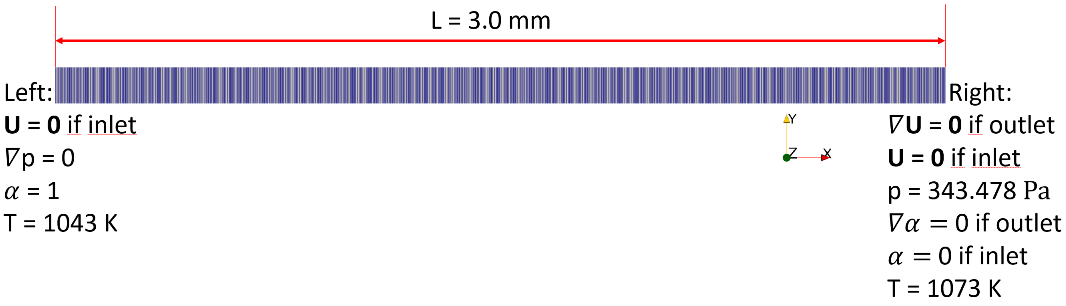

Figure 4.6: Mesh and boundary conditions

4.2.2 Results

The position of the interface as a function of time is analytically given by the following expression (see [2] for details):

$$\delta(t) = \sqrt{2t\,\alpha_{D,L}\left(\frac{1}{2}+\frac{H_{fg}}{c_{p,L}\,(T_{sat}-T_w)}\right)^{-1}} \tag{4.4}$$

where $\alpha_{D,L}$ is the liquid thermal diffusivity.

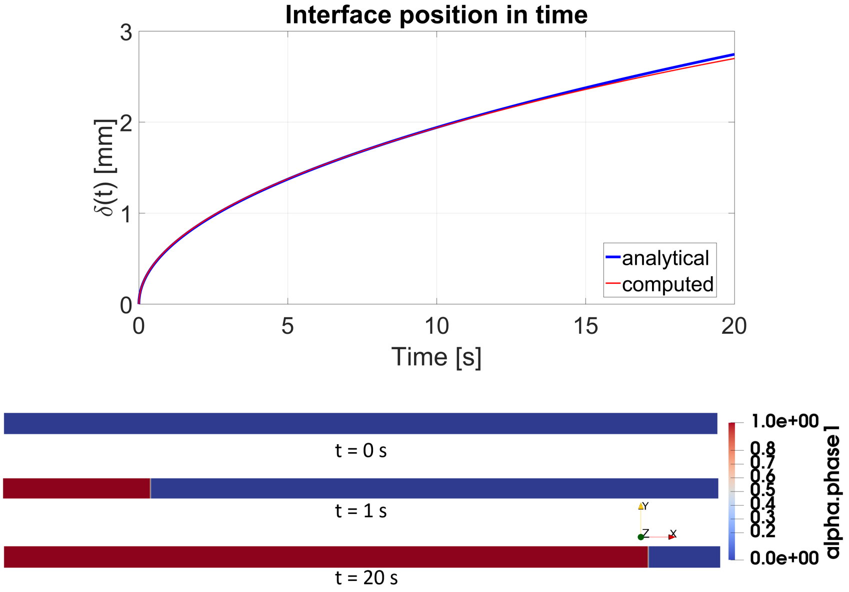

Figure 4.7: On the top, the position of the interface as a function of time. On the bottom, the liquid volume fraction inside the domain at three different times.

The top plot of Figure 4.7 compares the interface position computed through equation 4.4 and the results of the simulation. Although for large times $(t > 15\ s)$

the gap between the two curves seems to increase, there is still a very satisfactory overlap. It can be also noticed that the speed at which the interface moves is much smaller than that of evaporation; for instance, $\delta_{anal}^{evap}(20\ s) = 92.2\ mm$ while $\delta_{anal}^{cond}(20\ s) = 2.7\ mm$. Since, as it has been already anticipated, the computational cost that characterizes condensation case studies is higher than before, the length of the domain is assumed to be $L = 3.0\ mm$ and not $1.0\ m$. Indeed, the interface moves few millimeters in the time interval under investigation.

The bottom plot of Figure 4.7 shows the liquid volume fraction inside the domain at three different times. The interface is well defined and sharp, as it is expected since VOF is used; this is due to the dense mesh, which is composed of 1000 cells and not of 250 as for evaporation.

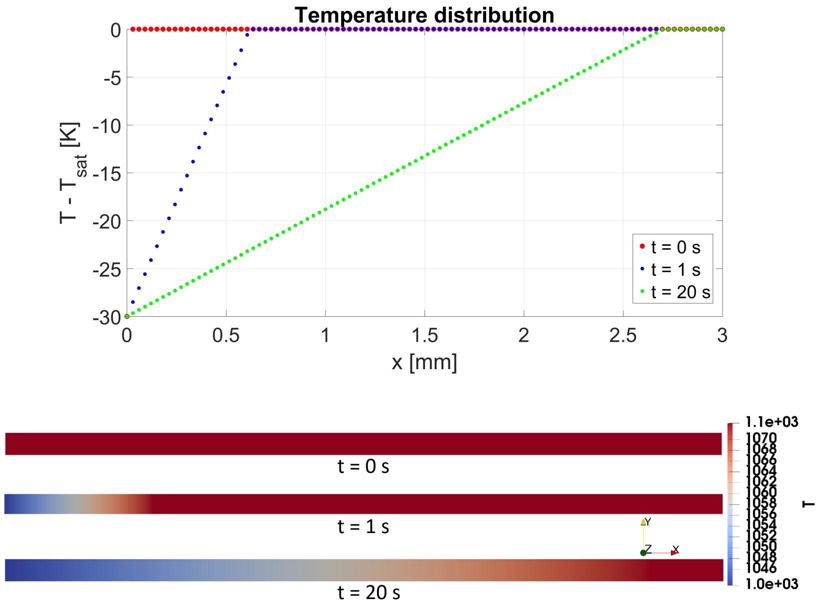

Figure 4.8: Temperature distribution inside the domain

Figure 4.8 shows the temperature distribution inside the domain at three different times. As it can be seen by looking at the left plot, the typical gaussian behavior is not caught; conversely, temperature linearly increases in the liquid phase, as it would happen in a diffusive steady state problem. The vapor phase is instead at saturation, as it should be. Since the liquid thermal diffusivity is ~ 2.7 times smaller than the vapor one (i.e. they have the same order of magnitude), the results are believed to be wrong and can be caused by one of the following reasons:

1. working fluid. In order to verify this hypothesis, another simulation with saturated water at $p = p_{sat} = 10\ bar$ and $T = T_{sat} = 453\ K$ has been performed; however, the results are qualitative the same and this means the problem is not in the fluid properties;

2. boundary conditions. Many different trials have been done, but no interesting conclusions have been reached;
3. evaluation of the phase change. To the opinion of the author, this is the main candidate explaining a similar behavior; indeed, condensation is much more complicated than boiling from the numerical point of view and codes based on VOF methods suffer a lack of stability (as it has been discussed in the previous chapter).

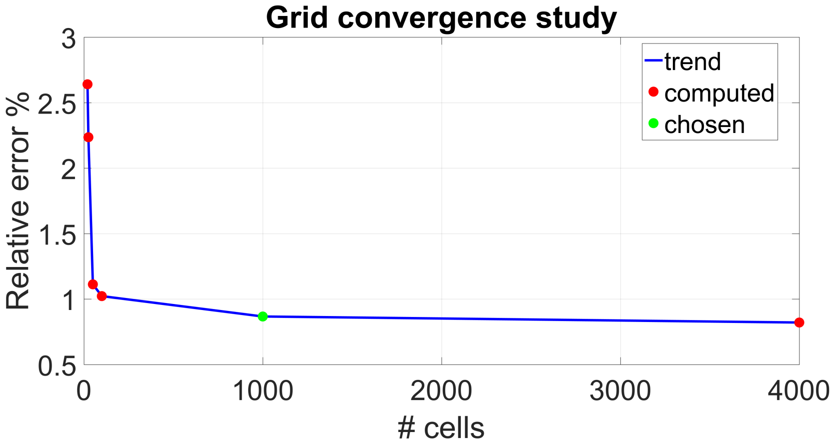

Figure 4.9: Grid independence study

The grid convergence analysis is shown above.

4.3 Nusselt laminar film

The Nusselt laminar vertical film problem is a typical 2D benchmark problem for codes evaluating condensation. The problem has indeed been studied by Nusselt and a semi-empirical treatment is available in literature (see [3]). A vertical wall at $T_w < T_{sat}$ transfers heat through conduction to the stagnant vapor, that is at $T = T_{sat}$; as a result, phase change occurs. Due to the effect of gravity, the produced liquid flows down towards the outlet, while new vapor enters the domain from the free surfaces, replacing any outgoing flux. The cold wall is covered by a film of liquid, which becomes thicker as it approaches the outlet, because at a fixed height there is not only the fluid flowing down from the upper part, but also the condensing vapor which is at that height. Reynolds number, whose expression for this particular case will be shown in the following, increases moving towards the outlet and this may determine a wavy or even a turbulent regime,to be avoided in the present simulation because the benchmark is conveniently carried on in laminar conditions.

4.3.1 Simulation setup

Figure 4.10 shows the mesh and the boundary conditions of the simulation.

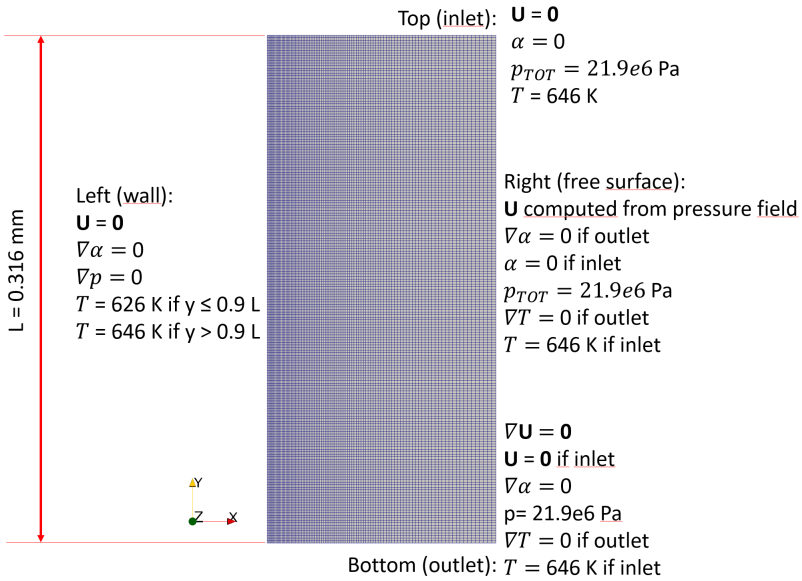

Figure 4.10: Mesh and boundary conditions

The domain, filled by saturated vapor at $t = 0\ s$, is made of 100x200 cells which are distributed so as to be refined close to the wall, where the liquid film is formed; indeed, the expansion factor along the x-direction is 4 (conversely, it is not used in the y-direction). This choice is clearly related to the physics which has to be simulated. An expansion factor smaller than 1 might be also suggested for the y-direction, since at the beginning of the film the interfacial curvature is high. This has been done but no big differences were found. Indeed, the work associated to the surface tension has not been implemented in the temperature equation, because in that case the code could fail to solve the coupled equations. As a consequence, the use of the expansion factor in the y-direction is avoided.

The height of the domain L is such that wavy conditions are never reached; in fact, it can be derived from [3] that:

$$\left.\begin{aligned} Re(\delta) &= \frac{4g\,\rho_L(\rho_L-\rho_V)\,\delta^3}{3\mu_L^2} \leq 30 \\ \delta(y) &= \left(\frac{4k_L\,\mu_L(T_{sat}-T_w)\,y}{g\,\rho_L(\rho_L-\rho_V)H_{fg}}\right)^{1/4} \end{aligned}\right\} \Rightarrow\ y^* = 0.284\,mm \ \Rightarrow\ L = \frac{10}{9}\,y^* = 0.316\,mm \tag{4.5}$$

where $y*$ represents the distance from the beginning of the film at which the transition from laminar to wavy occurs.

In the previous derivation, many important assumptions have been done:

1. wave-free laminar flow;
2. heat transfer at the interface can only occur by condensation and not by conduction from the vapor;
3. surface tension is zero;
4. thermophysical propertiesare constant. In particular, water at $p = p_{sat} = 219\ bar$ and $T = T_{sat} = 646\ K$ is used (the choice of this working fluid is explained below);
5. wall temperature at $20\ K$ below T_{sat}.

It can be noticed that $L > y$; this is a consequence of the way boundary conditions are set up. As it can be seen from Figure 4.10, the wall temperature is a step function whose value is T_{sat} if $y > 0.9\,L = 0.284\,mm$. Thanks to this choice, the liquid film starts inside the domain and not on the top boundary. This is believed to be beneficial from the numerical point of view; in fact, due to phase change, pressure inside the domain decreases and since on both the top and the right surfaces the velocity is computed from the pressure field (see [4] for details), there is a resulting ingoing mass flux which might affect the numerics of those cells where $0 < \alpha_L < 1$, that are already difficult to be treated as they are not divergence free.

Along the x-direction the domain is assumed to be $0.45\,L$. Thanks to the second equation of the system 4.5, it can in fact be estimated that, by using this value, the liquid film width at the bottom surface should be around 30% of the total one. On the one hand, this guarantees that no liquid could reach the free surface; on the other hand, the useless amount of vapor far from the film is minimized following the approach already proposed for the Stefan condensation problem.

The temperature BCs on the wall has already been discussed. As far as pressure and velocity are concerned, standard BCs are used (respectively, homogeneous Neumann and no slip) and so further comments are not necessary. Conversely, for the volume fraction the situation is different from that of section 4.1.1; in order to be consistent with the temperature, an initial film of liquid is set in those cells located in such a position that is $y < 0.284\,mm$ and $x < 2\,\mu m$. Such an inizialization is done in order to help the code when solving the equations at the beginning of the simulation. This expedient was not necessary in the 1D verification problems, which are simpler from the numerical point of view.

The bottom surface is treated as an outlet, so the already discussed *inletOutlet* BCs are used (it can be noticed that both $T = T_{sat}$ and $p = p_{sat}$ are of Dirichlet type).

The top surface is treated as an inlet. Since it is assumed to be a free surface, total pressure, which includes the potential and kinematic contributions, is fixed (details can be found in [4]); velocity, as it has anticipated, is computed from the pressure field; volume fraction is zero because the environment is supposed to be filled of saturated vapor; temperature is equal to T_{sat}.

The right surface is a free surface whose BCs are a mix between those of the inlet and outlet.

At this point, the choice of quasi-supercritical water must be discussed (thermophysical properties of both liquid and vapor are reported in Table 4.2).

Table 4.2: Thermophysical properties of liquid and vapor water

Quantity	Symbol	Value	Unit
saturation pressure	p_{sat}	$21.9e6$	Pa
saturation temperature	T_{sat}	646	K
vaporization latent heat	H_{fg}	$276.4e3$	J/kg
elastic gas coefficient	R	461.0	$J/(kg\,K)$
surface tension	σ	$0.07e-3$	N/m
liquid kinematic viscosity	ν_L	$1.16e-7$	m^2/s
liquid density	ρ_L	402.4	kg/m^3
liquid specific heat capacity	$c_{p,L}$	$2.18e5$	$J/(kg\,K)$
liquid thermal conductivity	k_L	0.545	$W/(m\,K)$
vapor kinematic viscosity	ν_V	$1.92e-7$	m^2/s
vapor density	ρ_V	242.7	kg/m^3
vapor specific heat capacity	$c_{p,V}$	$3.52e5$	$J/(kg\,K)$
vapor thermal conductivity	k_V	0.538	$W/(m\,K)$

As it is well explained in [5], at the interface between two fluids there are *capillary waves* whose propagation can impose a strict constraint on the numerical time step used in the simulation. These waves significantly affect the stability of the interface and are caused by perturbations caused by either forces, such as surface tension, the finite accuracy of the algorithm, or parasitic currents (see chapter 3).

In order to avoid stability issues, the following criteria must be adopted:

$$\Delta t = min\left(\frac{Co\,\Delta x}{|\vec{u}|},\ \sqrt{\frac{\rho_{min}\Delta x^3}{4\pi\sigma}}\right) \tag{4.6}$$

The first expression refers to the CFL condition (Co is the Courant number), while the latter accounts for capillary waves. The value ρ_{min} density should be the sum of the fluids densities; however, the minimum value, $\rho_{min} = min\left(\rho_L, \rho_V\right)$ is taken in order to be conservative. In both cases, Δx is the mesh size; clearly, if expansion factors are different from one (as in the present simulation), its minimum value should be taken. It can also be noticed that the CFL condition goes as $\propto \Delta x$, while the capillary waves criterion goes as $\propto \Delta x^{3/2}$; therefore, for a sufficiently refined mesh the second condition becomes the limiting one. Indeed, a very important conclusion can be derived: a coarse mesh gives less accurate results than those computed with a finer mesh; however, the code is more stable and the numerical instabilities that may arise could dramatically affect the accuracy. In other words, it is not possible to say *a priori* if a coarser mesh would give better results or not.

From Tables 4.1 and 4.2 it can be estimated that:

$$\frac{\left(\sqrt{\dfrac{min\left(\rho_L,\ \rho_V\right)}{4\pi\sigma}}\right)_{Li}}{\left(\sqrt{\dfrac{min\left(\rho_L,\ \rho_V\right)}{4\pi\sigma}}\right)_{H_2O}} = \frac{8.575e-3}{525.268} = 1.63e-5 \tag{4.7}$$

If the density summation is used instead of the minimum (as it seen done above), the resulting ratio would be $\sim 10^{-3}$. In both cases, it is clear that using lithium as a working fluid would require a very small time step. That is why quasi supercritical water is used. In fact, at this stage the aim is that of verifying how the code deals with phase change.

Indeed, different cases have been performed, both with lithium and water at standard conditions (which is between the previous working working fluids in terms of difference between the liquid and vapor phase properties). It has been found that results are inaccurate and the code is not stable, unless very small time step are employed. As a matter of fact, the use of lithium in the previous 100x200 mesh leads to a $\Delta t_{max} \sim 1\ ns$; this implies that reaching the final time $T = 1\ s$ takes 14 days running the simulation in parallel on 12 cores on computer «Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz» with RAM 64 GB .

4.3.2 Results

Figure 4.11 shows the evolution in time of the liquid volume fraction. The initial film which is present at $t = 0\ s$ quickly increases in size and it is already significantly thicker at $t = 0.01\ s$. It can be noticed that near the top the amount of liquid is slightly higher than close to the outlet; a sort of wave is in fact flowing down and this explains a similar shape. It can also be seen that the interface feels an higher diffusion there: there are more white cells than elsewhere and this feature has a numerical nature, because it is strictly related to the way the interface is solved. As a matter of fact, instabilities occur and they also determine the slightly unphysical shape that can be looked at $t = 1\ s$; indeed, near the outlet the interface is vertical.

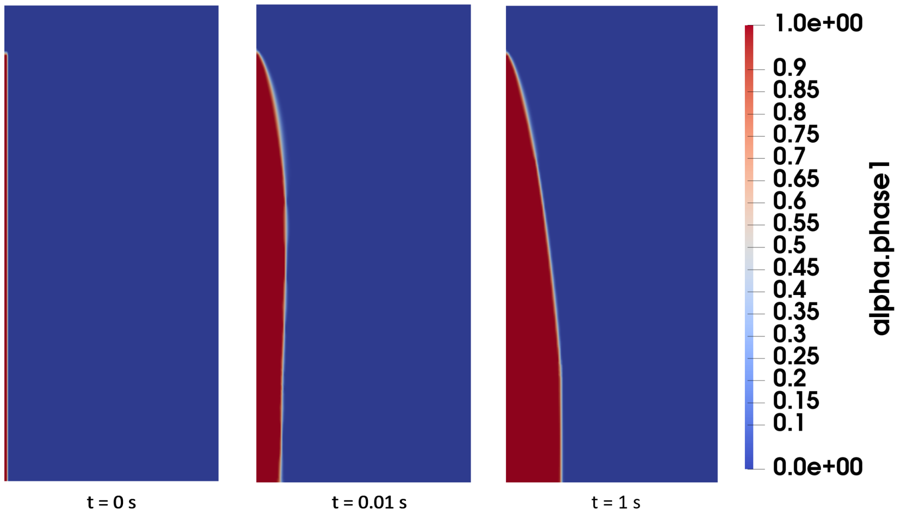

Figure 4.11: Liquid volume fraction distribution at three different times

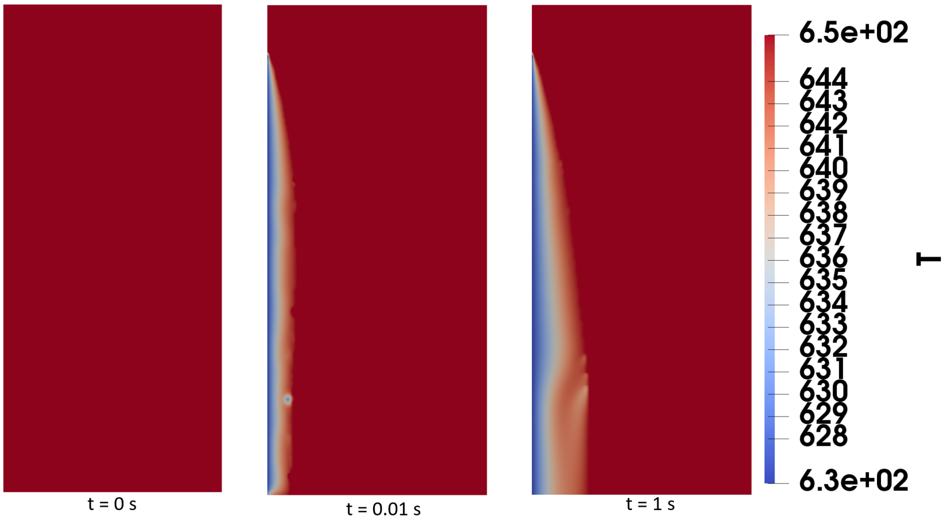

Figure 4.12: Temperature distribution at three different times

Figure 4.12 shows the temperature evolution inside the computational domain. At the initial time both the liquid and the vapor phases are at $T = T_{sat}$. However, because of the Dirichlet BC on the left wall, heat is removed and this results in a diffusive process through which the average temperature decreases. As it has been anticipated, numerical instabilities are present and they produce "cold spots" at some interfacial cells. They are not physical and can only be attributed to the numerics.

Indeed, cold spots also affect the temperature distribution inside the liquid film, as it can be noticed from the third figure (i.e. $t = 1\ s$). Obviously, the use of a more complicated fluid (i.e. lithium, whose thermophysical properties between phases are even more different than those of quasi supercritical water) may enhance stability issues, which can also affect the accuracy of the results.

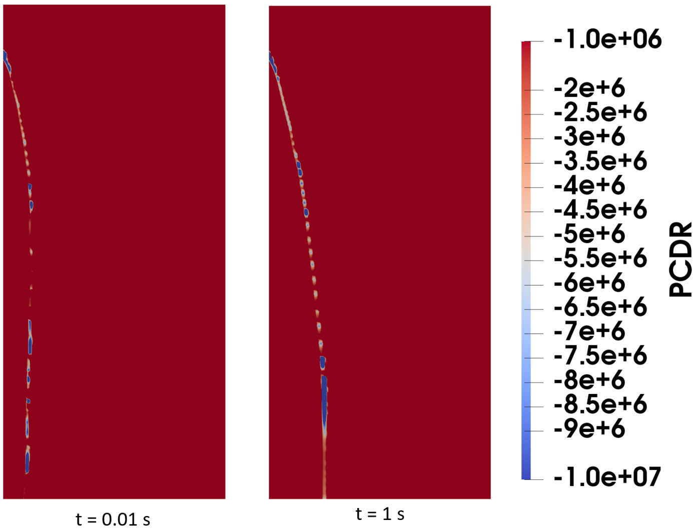

Figure 4.13: Phase change density rate [kg/(m^3 s)] distribution at $t = 0.01\ s$ and $t = 1\ s$

A further confirmation of these problems is given in Figure 4.13, which shows the volumetric mass transfer rate distribution at $t = 0.01\ s$ and $t = 1\ s$. The first thing to notice is that values are negative; this is only due to the way the post-processing has been performed and has no consequences. It must be also stressed that they are huge ($\sim 10^6$ kg/(m^3 s)); this is expected since the gradient of the liquid volume fraction $|\nabla\alpha_L|$ is significantly high at the interface.

However, according to the opinion of the author, the most important feature appearing in Figure 4.13 is that at those cells where cold spots are located the phase change density rate has its highest absolute values (indeed, they are of the order of 10^9 kg/(m^3 s)). This explains why the interface is vertical near the outlet: it follows from a bad resolution of the phase change.

The effect of surface tension at the top of the film, where the curvature is very high, should be also highlighted; even if a smoothing procedure has been implemented, it can enhance the issues already present and associated to the non divergence free feature which is typical of phase change flows.

The author believes that all these problems are due to:

1. the use of MULES: it does not account for non divergence free flows;
2. the reference solver: it is derived from a purely incompressible multiphase fluid dynamic solver and does not account for robust and stable phase change

procedures. As a consequence, both *interPhaseChangeFoam* and *interFoam* must be investigated.

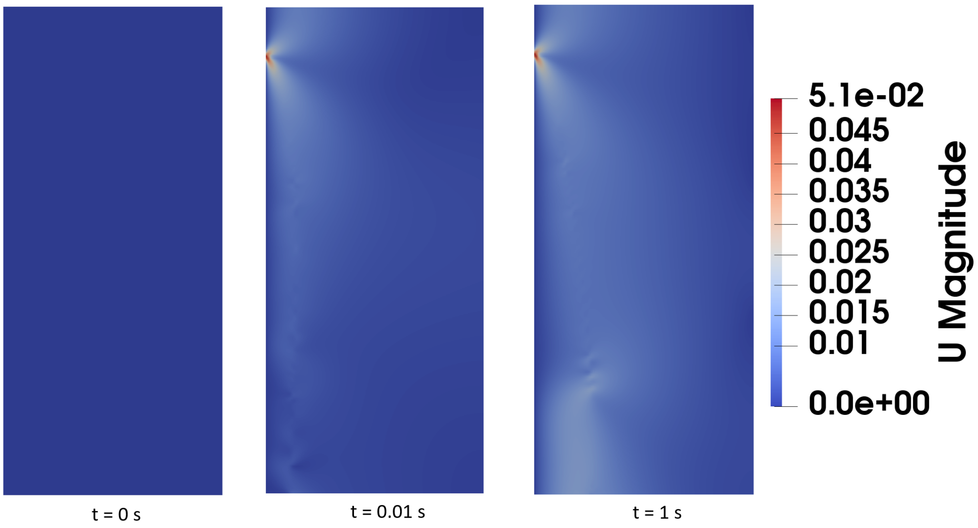

Figure 4.14: Velocity magnitude distribution at three different times

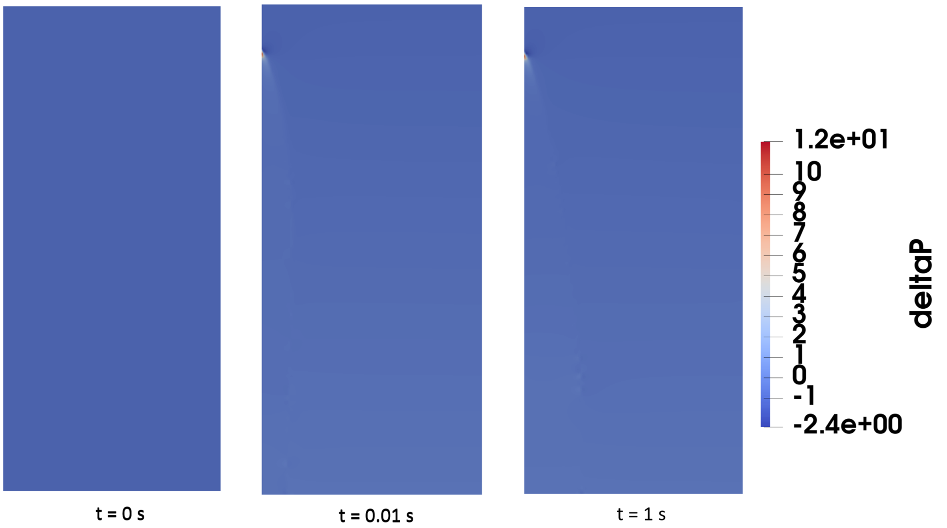

Figure 4.15: Pressure difference distribution at three different times

Figure 4.14 shows the velocity magnitude distribution inside the domain at three different times. The field distribution close to the top of the film is characterized by a spot. This explains why the simulation set up has been done in such a way that the film of liquid starts inside the domain; it can be noticed, in fact, that the velocity magnitude is very different from that of the inlet and this would result in a conflictive behavior which can lead the code to crash, if the present set up is not adopted.

Figure 4.15 shows the pressure difference distribution $p - p_{sat}$ inside the domain. There are not relevant gradients because the fluid is assumed to be stagnant; however, at the beginning of the film there is a spike, where numerical values are slightly higher than elsewhere. This feature is strictly related to the $p - U$ coupling resolution and is expected after the discussion on the velocity magnitude.

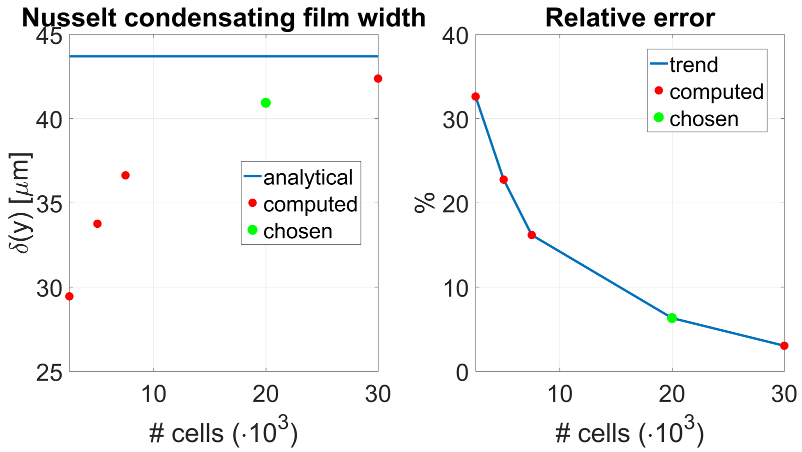

Figure 4.16: Grid independence study

Figure 4.16 shows the grid convergence analysis. Despite the issues associated to the resolution of the interface, the film width computed numerically approaches the analytical one as the mesh is refined. However, after a certain amount of cells (i.e. 100x300) capillary waves lead to very high instabilities; the amount of liquid at the outlet is significantly larger than before because of the effects of cold spikes, thus confirming that the mesh can not be refined as desired, since an upper limitation is imposed by the numerics.

4.3.3 Mass conservation

The verification of mass conservation is now performed.

The top plot of Figure 4.17 shows the time evolution of the liquid, vapor and total masses. Due to the small dimensions of the computational domain, the y-axis is in $\mu g/m$, since OpenFOAM only admits 3D meshes; therefore, results are divided by the length of the domain along the Z-direction.

The mass of liquid quickly increases from zero and reaches a steady state value that is $\sim 3.1\ \mu g/m$, whereas vapor decreases until $\sim 9.0\ \mu g/m$. The total mass in the domain increases, because the difference between the densities of the two phases is huge. However, the most important feature of the bottom plot is represented by the fact that a steady state value is reached after a short transient ($\sim 0.2\ s$).

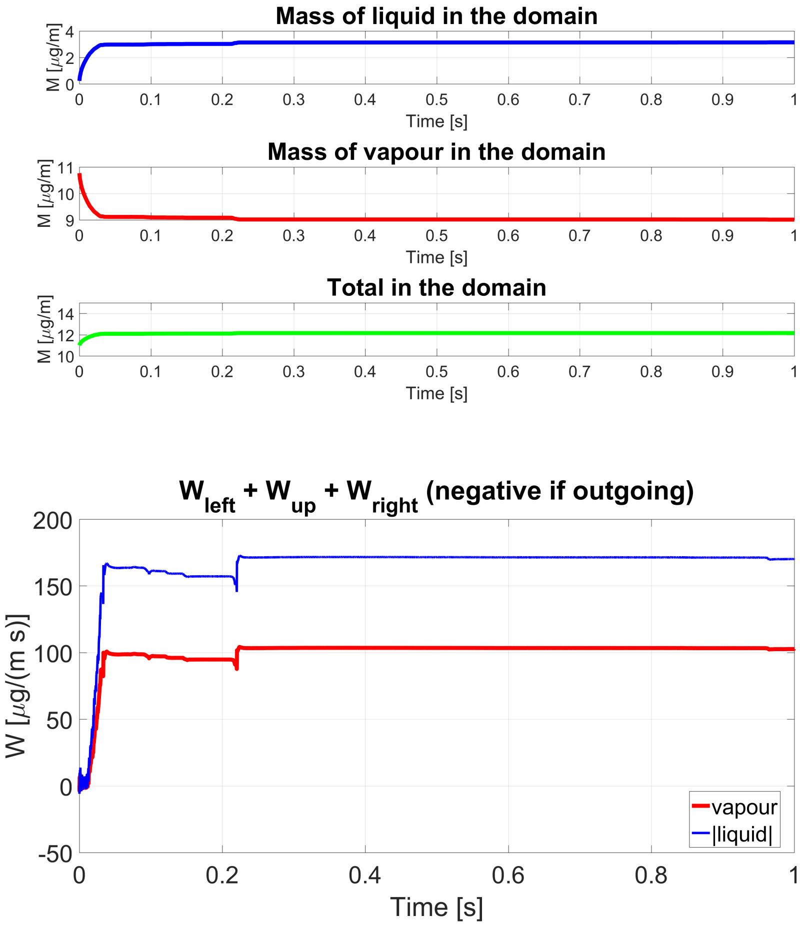

Figure 4.17: On the top, the evolution of the masses within the domain. On the bottom, the total flow rate of liquid and steam.

The bottom plot of Figure 4.17 shows the total mass flow rate of liquid and vapor, computed as the summation of the single flow rates over the different boundaries (clearly, the left wall is not taken into account). Ingoing fluxes are assumed to be positive; this convention is opposite to that used in OpenFOAM, because the scalar product between the vector velocity and the normal of the surface is negative in case of inlet.

Vapor flows inside the domain, while liquid exits from it; the absolute value of this phase, however, is the largest one and this implies that globally mass is flowing out. This result is opposite to what is shown in the bottom-left figure; in fact, according to the right plot it can be concluded that:

- steady state is never reached. The mass flow rate of liquid that is exiting from the bottom surface is, in absolute terms, larger than the ingoing mass flow rate of vapor. As a consequence, the total mass inside the domain should decrease, but this is opposite to what the green plot of the left Figure shows;

- the total mass within the domain decreases in time.

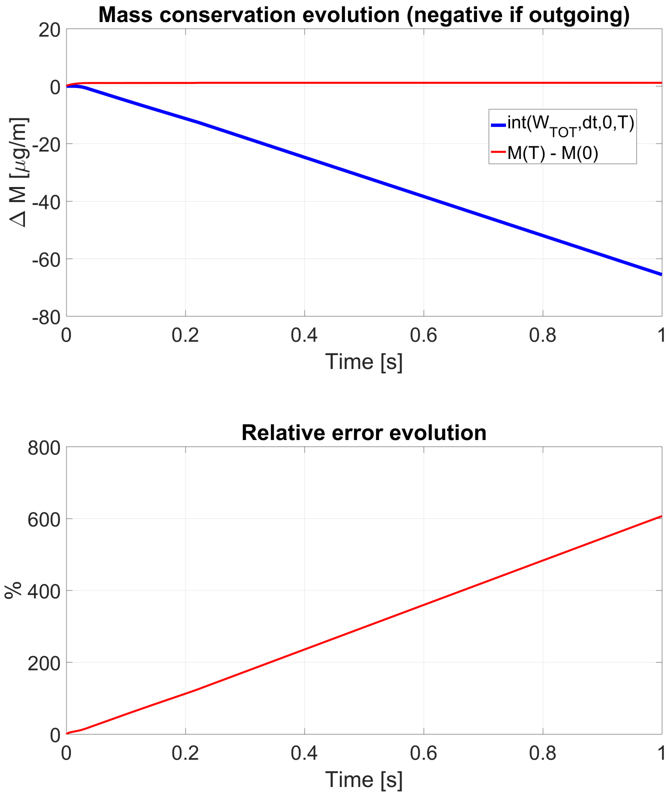

Figure 4.18: On the top, the evolution of mass conservation. On the bottom, the resulting relative error.

The top plot of Figure 4.18 further confirms the previous statements; the left and right hand sides (LHS and RHS) of the mass conservation equation are respectively computed as:

- $\Delta M_1 = M(t) - M(t=0)$. Data are those of the bottom-left plot of Figure 4.17;
- $\Delta M_2 = \int_0^t \left(W_{tot}^{liq} + W_{tot}^{vap} \right) dt'$. Data used to compute ΔM_2 are those of the right plot of Figure 4.17.

The bottom plot of Figure 4.17 shows instead the relative error, that is evaluated as:

$$err\% = 100 \frac{\Delta M_1 - \Delta M_2}{M(t=0)} \tag{4.8}$$

According to the opinion of the author, the most important feature of this plot is not related to the huge numerical values, which might be reasonable since $M(t = 0)$

is almost equal to the vapor mass, that is really small because its density is many order of magnitudes lower than that of liquid. The most relevant aspect is believed to be the fact that the error linearly increases in time.

Figure 4.19 shows the liquid and vapor flow rates on each boundary. As a matter of fact, liquid can only flow downwards through the outlet, because the external environment is assumed to be all filled by saturated vapor; moreover, the x-dimension of the domain is set up in such a way that the film never reaches it (see section 4.3.1)

Vapor can instead enter from the top and right surfaces, while it exits from the outlet because of the vertical drag force associated to the film motion, which falls down because of gravity.

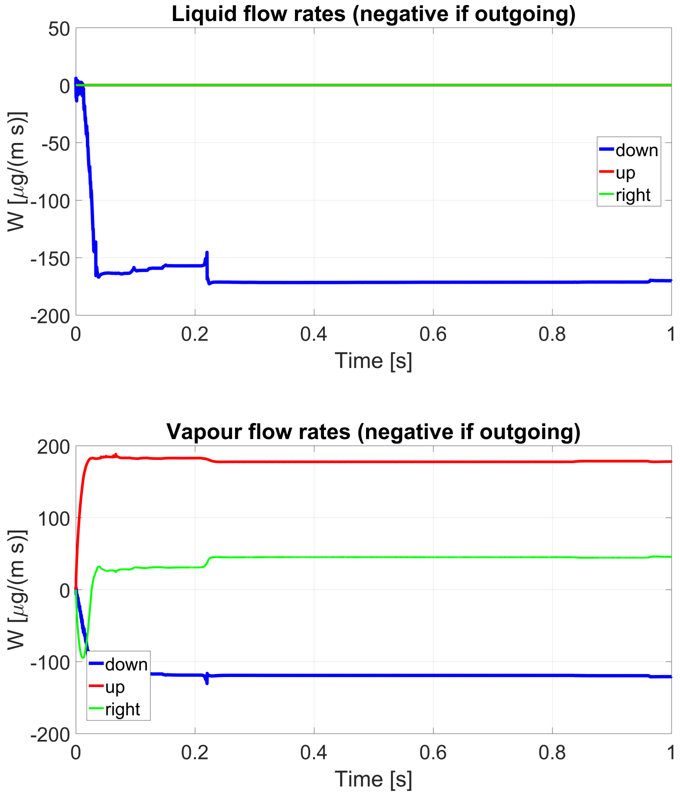

Figure 4.19: On the left, liquid mass flow rates along the different surfaces. On the right, vapor mass flow rates along the same surfaces.

At this point, it seems fundamental to discuss Figure 4.20, which shows the total volumetric flow rates of liquid and vapor. They are always equal and this let the author suppose that volumes, not masses, are conserved. Indeed, the reference solver, *interPhaseChangeFoam*, has been derived from another one, *interFoam*, which does not deal with phase change. Because of this, by looking at Figure 4.20 it seems very clear that the way interPhaseChangeFoam has been developed might be wrong; volume and mass are in fact proportional if and only if the one fluid is

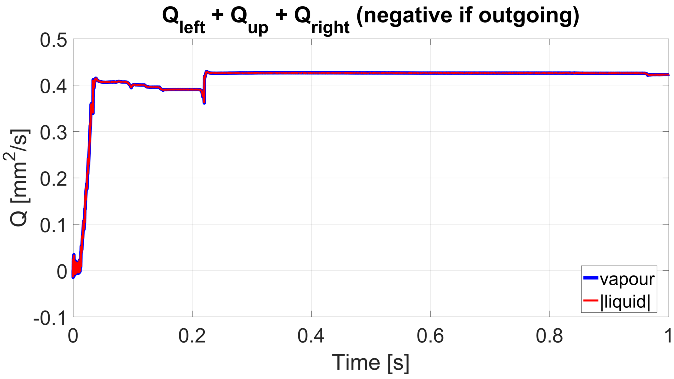

Figure 4.20: Total volumetric flow rates of liquid and steam per unit depth

divergence free. This is one of the assumption done in interFoam, but it is not true for the derived solver, because cavitation (and so phase change) occurs there.

As a matter of fact, in order to check these hypotheses, the same type of verification should be performed for both the solvers. This has been done running the reference tutorials released by OpenFOAM. After the analysis (not shown in the present thesis) has been carried out, it is found that:

1. mass conservation is verified in interFoam. The relative error reaches a steady state value which is $\sim 0.3\,\%$, meaning that:
   - the solver is well implemented;
   - the data run post-processing is set up properly.

   The tutorial used to perform the analysis is called *damBreak*;
2. mass conservation is NOT verified in interPhaseChangeFoam and the same features discussed in the Nusselt laminar film are qualitatively present: the relative error linearly increases in time (numerical values are around $10\,\%$ after $0.05\ s$). The tutorial adopted is called *cavitatingBullet*.

4.4 Conclusions

The verification of the numerical code has been performed and three different test cases have been studied.

Due to its simplicity, the evaporating 1D Stefan problem is firstly analyzed. The working fluid is lithium at saturation conditions and the BCs are such that a pure diffusive case is reproduced. Comparing the results obtained with the analytical formula that is available, a good match has been found.

Although the 1D condensating Stefan problem is the opposite situation compared to the previous one, the computation is much more difficult, as it can be noticed by looking at the mesh, that is much finer than before, and the computational cost, which is significantly higher. According to the author, the explanation of this feature is related to the α_L equation. During evaporation/boiling at $t = 0\ s$ the left hand side is different from zero (initially the domain is filled of liquid); conversely, during condensation the left hand side is zero (initially the domain is filled of vapor). As a consequence, the importance associated to the phase change term is much different and this affects the computation. Indeed, the shape of the temperature distribution at different times is not that expected, even if the analytical formula is well matched.

The Nusselt laminar vertical film problem is finally addressed. Both the verification and the grid independence studies are good, but several issues have been found:

1. depending on the fluid properties, stability constraints can be strict, leading for the case of Li to unaffordable low timesteps, at least for the computational resources available;
2. instabilities at the interface can dramatically influence both the code robustness and the accuracy of the results;
3. mass is not conserved if the solver myPhaseChangeFoam is used.

The causes of such problems have been analyzed and it has been found that the same issues are also present in the reference solver, that is interPhaseChangeFoam. Its implementation is questionable, as well as both the MULES algorithm and the way fluxes are computed in OpenFOAM.

References

[1] S.W.J. Welch et al. «A volume of fluid based method for fluid flows with phase change». 2000.

[2] N. Samkhaniani et al. «The evaluation of the diffusive interface method for phase change simulations using OpenFOAM». 2017.

[3] F.P. Incropera et al. *Funfamentals of heat and mass transfer*. 1996.

[4] «Boundary conditions - OpenFOAM-2.3.0». 2014.

[5] F. Denner et al. «Numerical time-step restrictions as a result of capillary waves». In: *Journal of computational physics* (2015).

Chapter 5

Liquid metal divertor

This chapter deals with the numerical simulation of a pool-type liquid metal vapor box divertor, that was the original objective of the present thesis. Obviously, because of all the difficulties and issues that have been discussed during the verification procedure, one can expect that the physical results will be bad. Nevertheless, this analysis is believed to be useful in order to point out some interesting features, such as how the mesh is produced, or how the bad solution of the interface affects the whole domain.

The way the domain is imported and meshed is firstly analyzed and a single phase flow simulation using one of the solver available in OpenFOAM is performed. The aim is to verify that no issues are related to the mesh. Then, a condensating test case, very similar to the 2D Nusselt laminar vertical film, is faced. Because the results of the CFD are absolutely not satisfactory, different ways to approach the problem of thermal phase change are proposed.

5.1 CAD import and mesh

It is chosen to produce the CAD geometry in STAR-CCM+, which is one of the most powerful commercial software dealing with CFD. Indeed, STAR-CCM+ is considered to be the best mesher available at the moment; the use of OpenFOAM is preferred to develop the physical model because, being open source, it gives the chance to manually implement the governing equations.

The output file obtained in STAR-CCM+ (.ccm format) must be converted into a format which is readable for OpenFOAM. To do that, an utility called *ccm26ToFoam* is used. In this way, the mesh can be produced in STAR-CCM+ and then imported in OpenFOAM.

At this point, in order to impose axisymmetric BCs on the front/back faces, the prismatic mesh should be further modified in order to obtain a wedge; this requires another utility, that is *extrudeMesh*. To make it work, it is mandatory that a surface of the prism coincides with the axis of symmetry of the wedge. Because of this, a fictitious domain (deleted at the end of the procedure) is needed.

Figure 5.1: CAD geometry of the divertor in STAR-CCM+

Figure 5.1 is useful to better clarify what explained above. The CAD represents a prism whose width along the z-axis is much smaller than those along the other directions. The top left corner coincides with the origin of the reference frame, so the left surface of the outer dummy parallelepiped is such that $x = 0$. After extrudeMesh is used, this patch becomes the axis of symmetry of the resulting wedge. Finally, everything but the divertor (that is purple in Figure 5.1) is deleted. Indeed, it might be seen from the *boundary* file available in *polyMesh* (this folder is automatically created in OpenFOAM after the mesh is produced) that the dummy figures all have boundary surfaces with no cell faces, meaning that they do not practically exist anymore (this further confirms why they can be safely deleted).

In principle, one might ask why such a complex procedure is required; it would be much easier to already build a wedge geometry in STAR-CCM+. However, according to the OpenFOAM *wedge* BCs implementation, the axis of symmetry must be one of the surface of the input CAD and this explains why the big dummy rectangle is mandatory. Furthermore, the resulting mesh may be topologically not correct because the front and back surfaces are not planar. To reduce as much as possible the amount of constraints coming from whatever is not part of the OpenFOAM package, the use of extrudeMesh is required; in this way the only input that is needed is in fact a prismatic CAD.

Table 5.1 shows the main input parameters of extrudeMesh, chosen based on the best practices suggested for this utility. For example, the axis point must be the origin of the reference frame, otherwise a crash occurs while running the utility. If this had not been the case, the use of the dummy rectangle would have been avoided. Similarly, the wedge angle must be $5.0°$; nLayers is equal to one and the

Table 5.1: extrudeMesh numerical input

Quantity	Symbol	Value
Extrusion model	extrudeModel	wedge
Flip surface normals	flipNormals	false
Number of layers	nLayers	1
Axis point	axisPt	$(0\,0\,0)$
Wedge angle	angle	5.0°

software extrudes ± 2.5° with respect to the reference plane.

Figure 5.2 shows the CAD and the final mesh in OpenFOAM of a outboard liquid metal divertor.

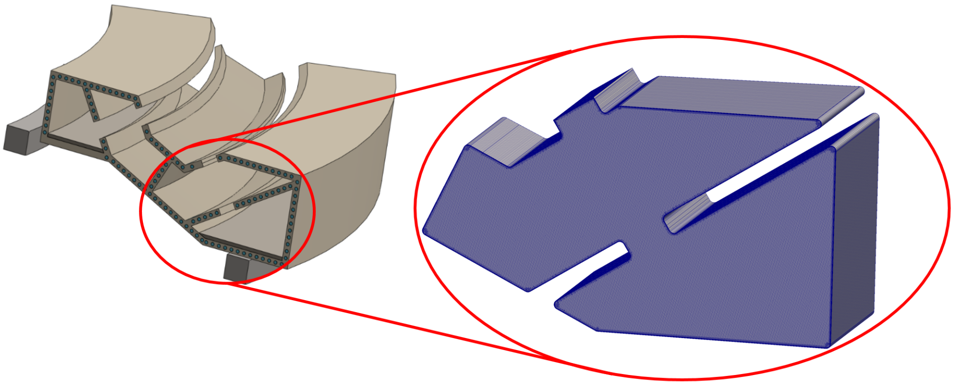

Figure 5.2: CAD and mesh of the divertor in OpenFOAM

To summarize, in order to get the desired mesh the following procedure must be adopted:

1. create a 3D prismatic geometry in STAR-CCM+, ensuring that one corner coincides with the origin of the reference frame;
2. import the mesh from STAR-CCM+ to OpenFOAM by applying the ccm26ToFoam utility;
3. apply the extrudeMesh utility in order to get a wedge from the prismatic geometry. Input parameters must be those of Table 5.1;
4. manually delete all the dummy geometries.

5.2 Mesh verification

In the previous section it has been explained how to generate the mesh; the outcome is topologically correct and valid. However, when performing a simulation it might happen that issues which are somehow related to the mesh could arise (this is particularly true when the geometry is complex, as it is in the case under analysis). Therefore, a mesh verification can be very useful to prove that nothing troublesome occurs because of the mesh.

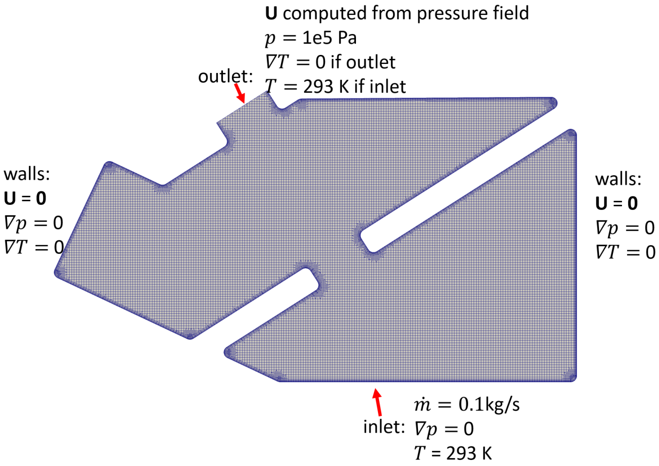

Figure 5.3: Mesh verification setup

A single phase flow simulation is performed: the OpenFOAM solver which is adopted is called *rhoPimpleFoam*, designed for transient turbulent flow of compressible fluids. The simulation set up is exactly equal to that of the tutorial *angledDuct*; the only difference is that no porosity walls are taken into account. The fluid is air and the RAS model is $k-\epsilon$. A schematic representation is shown in Figure 5.3.

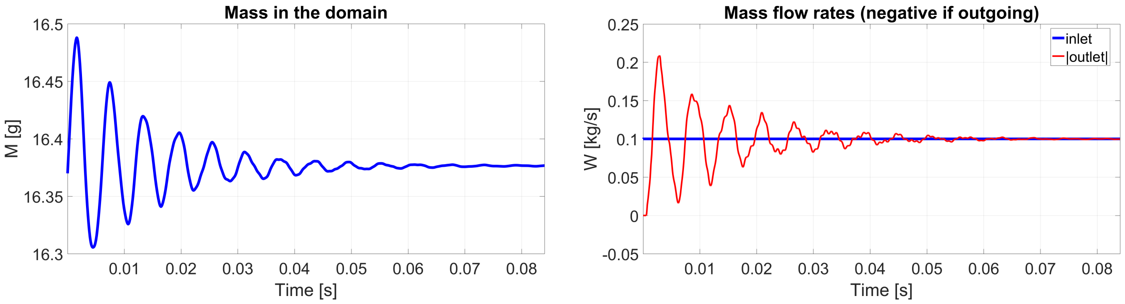

Figure 5.4: On the left, the mass evolution within the domain. On the right, the inlet and outlet flow rates.

Figure 5.4 shows the mass (left plot) and flow rates (right plot) evolutions in time. While the inlet flow rate is constant (this is due to the BC chosen for the velocity field), at the outlet oscillations occur, although they are progressively dumped out, until steady state is reached. This explains the mass evolution; in fact, at the beginning there is a net flux entering the domain, so the total mass increases. After that, because of the oscillating behavior, the amount of fluid exiting becomes higher than that entering, so the mass inside the domain decreases. Oscillations goes on until the steady state is reached. In other words, the left and right plots of Figure 5.4 perfectly agree each other.

Figure 5.5 shows the mass conservation; in particular, the left and right hand sides (LHS and RHS) are plotted on the left and are computed in the same way that has been discussed in the previous chapter. The resulting relative error is absolutely small and negligible and reach a constant value, so it does not increase in time (as it happens both for interPhaseChangeFoam and myPhaseChangeFoam).

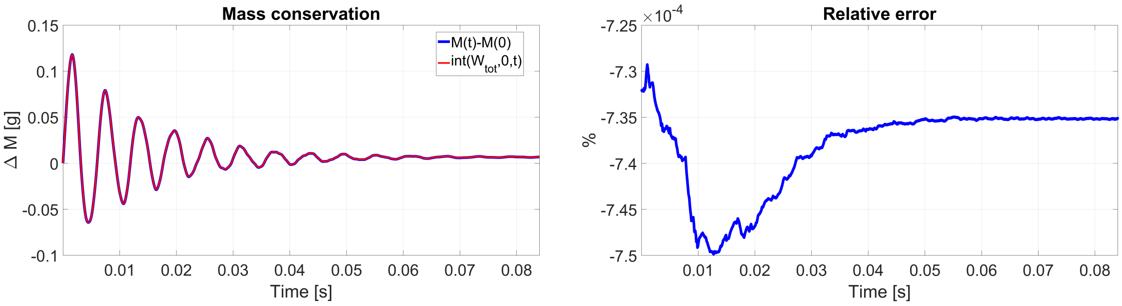

Figure 5.5: On the left, the evolution of mass conservation. On the right, the resulting relative error.

In conclusion, thanks to this benchmark it is proved not only that the mesh is topologically valid, but also that when running a simulation the mesh does not give rise to unexpected behaviors. The procedure to get it is therefore absolutely right.

5.3 Condensating simulation

The test case presented in this section is very similar to the Nusselt laminar vertical film, which has been discussed in the previous chapter. The domain is all filled by saturated vapor of quasi supercritical water, whose thermophysical properties are exactly equal to those already analyzed in the previous chapter. The walls are at a temperature $T_w < T_{sat}$. There are only two differences with respect to Nusselt problem:

1. $\alpha_L = 1$ at the walls. This implies that the initialization consists in a pool of liquid that is located at the bottom of the domain. No initial film of liquid is near at the walls;

2. A single outlet is present. This reflects the physical situation which is desirable to reproduce: mass can be lost from the domain only through the nozzle between the DC and the MC.

Figure 5.6 summarizes the simulation setup.

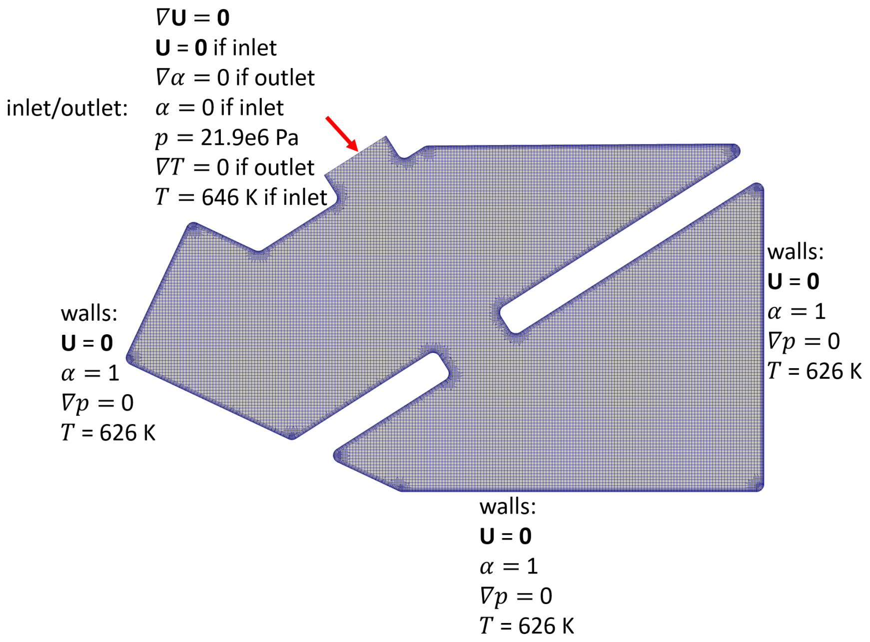

Figure 5.6: Simulation setup

5.3.1 Results

Figure 5.7 shows the liquid volume fraction at three different times. At $t = 1\,ms$ the condensed accumulate at the top left corner, as if a droplet is going to be produced. Moreover, it can be noticed that the liquid-vapor interface is not smooth and this reflects the fact that phase change is occurring there. However, by looking at the rightmost figure it is clear that the resolution is not physical; in fact, the liquid that is produced at the top wall through condensation does not fall because of gravity. Furthermore, unphysical vapor spots are located near the sharpest corners. These spots derive from the dynamic of the condensation process: the liquid produced at the walls become thicker and thicker, untill it intersects the liqud-vapor interface, which is increasing as well. When the interaction happens, the vapor spots are produced and remain there.

It can also be noticed that the volume fraction computed at the nozzle is troublesome; because of condensation there is a pressure reduction inside the domain, producing an ingoing flux which is non physical.

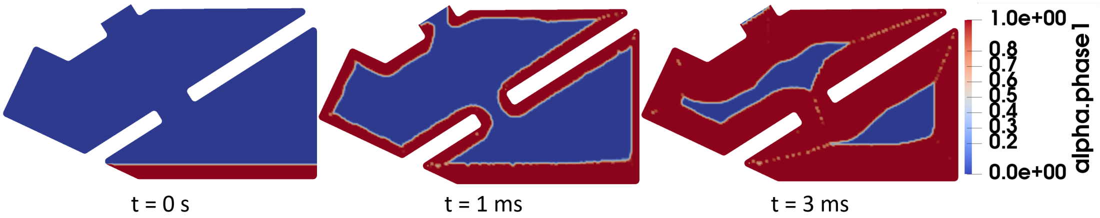

Figure 5.7: Liquid volume fraction distribution at three different times

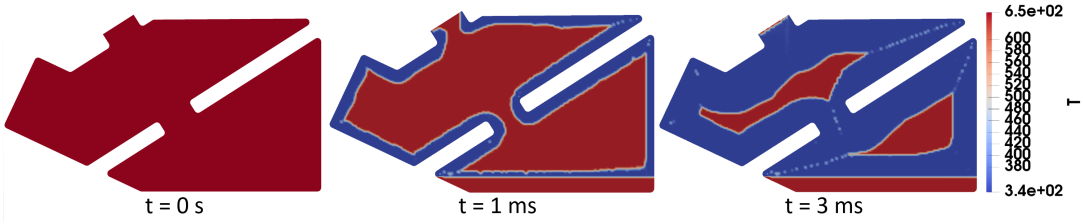

Figure 5.8: Temperature distribution at three different times

Figure 5.8 shows the temperature distribution inside the domain. It is absolutely not physical, because the computed minimum value is much smaller than T_w, which is the lower bound that should be reached.

As a matter of fact, such a feature reflects the main problem of the numerical code, that is the resolution of the moving interface. In fact, equations are not correctly solved only in those computational cells which it overcrosses.

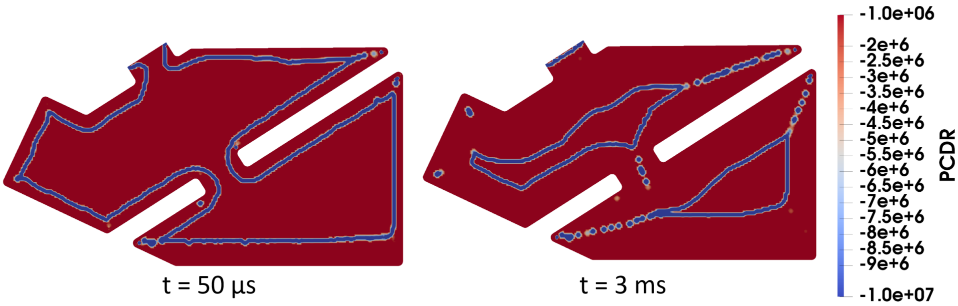

Figure 5.9: Phase change density rate [kg/(m^3 s)] distribution at $t = 1\,ms$ and $t = 3\,ms$

Figure 5.9 shows the volumetric mass transfer rate at $t = 1\,ms$ and $t = 3\,ms$, further confirming what has been explained. In particular, it is interesting to highlight that the phase change density rate is not zero (as it should be) where vapor spots are present. It should also be mentioned that numerical values are negative. As in the case of the Nusselt laminar film, this is only due to the way the post-processing has been performed and has no consequences. Indeed, by comparing Figures 5.9 and 5.8 it is clear that the main problem of the numerical code is the way the interface is solved.

5.4 Other approaches to the simulation of a LM vapor box divertor of pool type

In order to solve the issues that have been analyzed, a new algorithm dealing with the interface that is different from the MULES should be implemented. This was already attempted by Kunkelmann and Guedon ([1] and [2]), who both uses available numerical procedures such as the Piesewise Linear Interface Capturing (PLIC) method, but no relevant results have been achieved. It appears that a completely new algorithm is required and this goes beyond the current knowledge of the author and is incompatible with the limited time available for a thesis. Some trials can been found in literature (see [3]), but they are still under investigation.

Because of this, new approaches are investigated. In particular, the author focused on two specific roads:

1. *Euler-Euler*: the OpenFOAM reference solver is called *reactingTwoPhaseEulerFoam*. Since it solves two sets of conservation equations for each phase, this code could potentially solve the issues associated to the VOF approach; moreover, a mass fraction equation is included to also take into account chemical reactions. However, interfacial phenomena must be carefully modeled and this might not be straightforward; furthermore, by default no models associated to condensation is implemented, while there are some for boiling. To manually implement a condensation model accounting for the physics of droplets is beyond the knowledge of the author, since the informatic structure of the code is much more complicated than that of VOF solver. As a consequence, this road has been undertaken but has not lead to any relevant results;

2. *Finite Elements Methods* (FEM): since the main problem is associated to the non conservation of mass at the interface, a weak formulation can be adopted to overcome the difficulties. Indeed, Finite Volume Methods (FVM) write the conservation equations in a strong form, thus imposing them in each cell of the computational domain. Conversely, a FEM approach only guarantees global balances, so it relaxes the constraints associated to the single cells, such as those of the interface. This road is actually under study by mean of the commercial software Comsol Multiphysics 5.0, since it is modular and user friendly. The main idea is to couple the levelSet module, which deals with two phase flows without phase change (see [4] for details), with the thermal module, which accounts for energy balance; indeed, some coding is required also in this case, since the default continuity and level set equations do not have any source/sink terms, as it should be for condensation/boiling.

5.5 Conclusions

The numerical simulation of condensation in a pool-type liquid metal divertor has been unsuccessfully attempted.

The mesh is produced in STAR-CCM+ and then imported in OpenFOAM. The procedure is explained, putting some emphasis on the way the wedge geometry can be obtained from a prism.

A benchmark is then presented in order not only to show that the mesh is topologically correct, but also to prove that no hidden issues related to the geometry are present.

A condensating simulation, physically similar to the Nusselt vertical film, is finally performed. The same problems explained in the previous chapter have been found. This goes in the direction of implementing completely new interface reconstructing algorithm, different from MULES of PLIC, which is required to deal with an interface where thermal phase change is present. This goes beyond the knowledge of the author; indeed, some trials are present in literature (see [3]), but they are still under analysis.

In order to address the 2D modeling of the box type liquid metal divertor, other different roads have been investigated (i.e. Euler-Euler approach and Finite Elements). In the end, it was found that the best strategy is represented by a molecular study which exploits the Direct Simulation Monte Carlo methods. This will be the object of the next part of the thesis.

References

[1] C. Kunkelmann. «Numerical modeling and investigation of boiling phenomena». 2016.

[2] G. Guedon. «Two-phase heat and mass transfer modeling: flexible numerical methods for energy engineering analyses». 2013.

[3] L. Malan. «Direct numerical simulations of free surface and interfacial flow using the VOF method: cavitating bubble clouds and phase change». 2016.

[4] Comsol Multiphysics. *CFD module users's guide*. 2015.

Part II

Direct Simulation Monte Carlo evaluation of metal vapor dynamics

Chapter 1

Introduction: Part II

As it has been anticipated in chapter 1.1.1, the reference engineering model of a box pool-type divertor, discussed in [1], employs of a 0D model to evaluate the thermodynamic state of the LM and the vapor inside the chambers. This is particularly suitable to perform fast computations, but it is troublesome whenever a detailed analysis is required, since local thermal hotspots are not evaluated. Features such as temperature, pressure and density distributions are in fact not evaluated. The latter is particularly relevant for engineering purposes, because it is proportional to the amount of radiation that is produced from the interactions between the evaporated/sputtered LM ions and the incoming plasma flux [2]. In the following, this last statement will be clarified.

A general formula accounting for radiation losses is suggested in [3]:

$$P_R = n_e \, n_Z \, L_Z \left(T \right) \tag{1.1}$$

P_R is the radiative power per unit volume, expressed in [W/m^3]; n_e and n_Z are respectively the electron and impurity ions densities (units are [particles/m^3]). $L_Z \left(T \right)$ is the *loss function* and takes into account different contributions:

- *line radiation*: due to the collisions with a plasma particle, an impurity atom is excited. The successive de-excitation process of its electrons is such that a photon is emitted;
- *ionization*: during a plasma-vapor interaction, the energy transfer can be such that ionizations can occur;
- *Bremmstrahlung radiation*: charged particles undergo Coulomb collisions, that causes the emission of a gamma ray to guarantee momentum conservation;
- *other type of radiations*: an example (not discussed here) is the «cyclotron radiation». Details are available in [4].

As far as a LM vapor box divertors are concerned, n_Z is equal to the vapor density.

From the foregoing discussion, it is clear that it is desirable to have a density as high as possible in the EC to enhance vapor-plasma interactions and therefore plasma cooling. Indeed, if plasma loses energy due to radiation, this energy will be deposited on the box walls rather than on the small plasma wetted area. If, instead, energy is lost due to ionization, this energy will be released where recombination occurs. The latter location is the pool itself, for an attached plasma, and the vapor for a detached plasma. In the best scenario, then, all the power loss from the plasma will contribute to the decrease of the power deposited on the target via the advection/conduction channel, which is highly anisotropic.

The 2D model of the vapor, that will be presented in the next chapter, is going to be applied to the geometry described in section 5.1 and its results will be compared with those obtained via the currently available 0D model. Therefore, a summary of the main features of the latter is given in the following section, that is based on [5] and [2].

1.1 0D thermodynamic model

Figure 1.1 shows a schematic view of the pool type vapor box LM divertor and the most relevant physical phenomena.

The 0D model solves mass and energy conservation equations of Li atoms for two different control volumes, that are the DC and the EC (left part of Figure 1.1). The main assumptions are:

1. the Li pool instantaneously collects the re-condensed LM on both the EC and the DC walls;
2. the Li atoms mass flow rate going from the DC to the MC is exactly compensated by an equal amount of replenishing liquid Li supplied in the pool;
3. vapor is entrained (i.e. ionized) by the plasma, but recombines before reaching the pool if plasma is found to be detached by the 1D SOL model;
4. Li liquid is optically thick, while vapor is not;
5. vapor and liquid are in thermodynamic equilibrium in the EC.

As a consequence, when writing the mass balance of Li atoms at the EC (that includes both liquid and vapor), the variation in time is equal to the algebraic sum of the following source/sink terms:

1. *net condensation rate at the DC*: only the contribution coming from the DC is relevant (first hypothesis). In fact, inside the EC Li can change phase, but this would not represent a source nor a sink for the control volume of the EC;

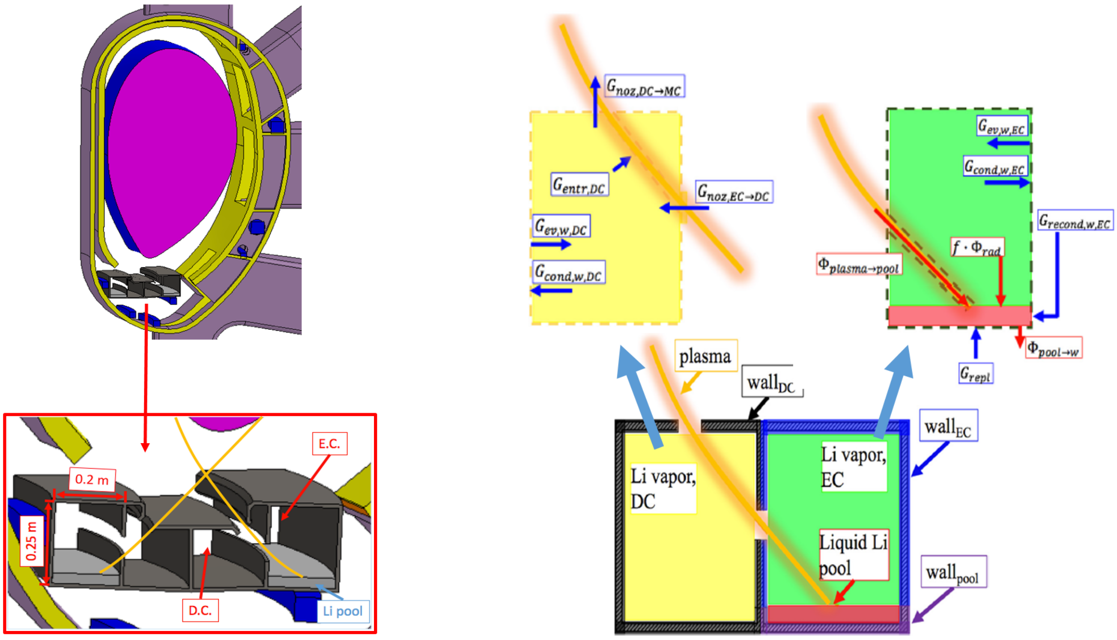

Figure 1.1: On the left, schematic view of the DTT plasma chamber with the pool type vapor box LM divertor highlighted. On the right, a schematic of the control volumes and respective fluxes for conservation of mass and energy (adapted from [5]).

2. *replenishment rate*: this term directly comes from the second assumption;
3. *entrainment rates*: interacting with the plasma, which moves at the sound speed, Li atoms ionize. As a consequence, these ions start moving towards the Li pool. They can recombine within the vapor phase (detached plasma) or at the pool surface (attached plasma), but in both cases they will represent a source for the mass balance of the EC. Three additional source rates must therefore be taken into account: entrained particles from the DC, the nozzle between EC and DC and the EC;
4. *flow rate towards the nozzle*: it is the atom mass flow rate going from the EC to the DC;

As far as the DC is concerned, the mass variation in time is due to:

1. *net condensation rate at the DC*, *entrainment rate at the DC* and *flow rate towards the nozzle*: they are equal to those analyzed for the EC, but have opposite signs;
2. *flow rate towards the MC*: it is the atom mass flow rate going from the DC to the MC.

Energy conservation equations account for the energy fluxes associated to the mass fluxes. One additional source term is added to the balance at the EC, that is the fraction of radiation power impinging the pool, that is optically thick (four hypothesis).

1.1.1 Comments

It seems relevant to point it out that a 0D model cannot evaluate spatial gradients; indeed, thermodynamic variables are assumed to be «step» functions with constant values in each chamber. However, the jump associated to such a discontinuity, if large enough, can determine the so called *chocked flow* condition, that is reached when the following holds:

$$\frac{p_{DC}}{p_{EC}} \leq \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma+1}} \tag{1.2}$$

where p is the pressure, while γ is the isentropic coefficient (it is equal to 5/3 for monoatomic gases, such as Li vapor).

Pressure, temperature and density jumps can be such that supersonic conditions are reached. This would potentially results in a shock whose wave front has to be identified, because thermodynamic variables change significantly there. Moreover, the shape of the nozzle (convergent, divergent, «neutral») may be optimized to control the position of the shock. A 2D analysis is therefore required to catch the pressure, temperature and density distributions inside the EC and the DC.

Some other considerations are also provided:

- the 0D model assumes thermodynamic equilibrium between liquid and vapor, so it is $p_{EC} = p_{sat}$. This is true close to the pool, but it is over conservative close to the other walls, which work as absorbers. The further from the pool, the lower the amount of vapor particles and, consequently, the value of pressure. As a result, it is expected that the 0D model overestimates this thermodynamic variable;

- the evaporating and condensating mass fluxes are evaluated through the Hertz-Knudsen formula. The same physical modeling of phase change will be also used in the 2D model to describe the interactions with the condensing/evaporating walls. In this way, the results of the two approaches can be fairly compared.

1.2 Conclusions

The 0D model that is currently used in the reference engineering model of a vapor box divertor can perform fast computations, accounting for the most important physical phenomena that are present. However, it cannot evaluate temperature, pressure and density distributions, which can be very important to properly determine the effect of the radiation losses, as well as the presence of shocks and the distribution of the condensation mass flow rate on the walls. As a consequence, a 2D modeling is required for more detailed studies.

References

[1] G.F. Nallo et al. «Development of a self consistent model for determining engineering parameters of a closed LM divertor». 2017.

[2] G.F. Nallo. «Liquid metal divertor modeling for future fusion reactors». 2016.

[3] P.C. Stangeby. «The plasma boundary of magnetic fusion devices». 2000.

[4] J.P. Friedberg. «Plasma Physics and Fusion Energy». 2007.

[5] G.F. Nallo et al. «Modeling the lithium loop in a liquid metal pool-type divertor». In: *Fusion Engineering and Design* (2017).

Chapter 2

Physical model

Inside the EC, part of the LM boils because of the interaction with the incoming plasma flux. The large evaporation rate which follows may therefore determine a relatively large vapor density, not withstanding the sinks. Therefore, the fluid might still be treated as a continuum inside this chamber (this has been discussed in chapter 2.1). Conversely, the vapor density inside the DC is expected to be really low. Indeed, differential pumping is employed for the purpose of decreasing as much as possible the probability of having plasma contamination inside the MC. As a result, the continuum hypothesis is not applicable for the DC and molecular simulation methods are required to simulate the metal behavior inside the divertor. In the following, these qualitative reasoning will be supported by quantitative assessment based on the available models.

First, the Knudsen number is evaluated in order to quantify how far the metal is from a continuous fluid. Then, the DSMC method is discussed. Finally, procedures to properly choose the computational cell dimensions, the timestep and the scaling factors required to perform the simulation are explained.

2.1 Evaluation of the Knudsen number

The Knudsen number is a dimensionless quantity giving the degree of rarefaction of a fluid. It is defined as:

$$Kn = \frac{\lambda}{L} \tag{2.1}$$

where λ is the mean free path of a generic particle, L is a properly defined characteristic length scale, which will be better discussed in the following.

As a matter of fact, having $Kn << 1$ means that a particle travels a distance much smaller than L between two successive collisions. This implies that the fluid is sufficiently dense to be treated a continuum. On the other hand, if $\lambda \sim L$, the fluid is rarefied and should be treated using molecular approaches.

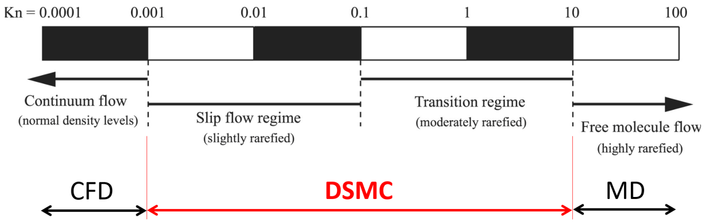

Figure 2.1: Molecular regimes identified by the Knudsen number. Adapted from [1]

Figure 2.1 is taken from [1] and shows the different regimes that can be identified based on the Knudsen number. These regimes are briefly described below, together with a mention of the appropriate modeling tool for each of them:

- *continuum*: intermolecular collisions dominate the particle motion and the whole fluid can be described by the Navier-Stokes equation;
- *slip flow*: the fluid is rarefied, but Navier-Stokes equations can be still applied, provided that the no slip condition at the walls is relaxed;
- *transition*: Navier-Stokes equations do not represent an adequate model. Boltzmann equation is used. Nevertheless, the amount of particles constituting the system is still sufficient to deal with collisions using statistical approaches. This will be better clarified in the following section;
- *free-molecular*: the flow extremely rarefied, particles rarely interact with each other.

A detailed description of each regime is available in [1].

In order to determine the collisionality of the metal vapor inside the box type divertor, the following analysis is provided.

The left plot of Figure 2.2 shows the time evolution of the Knudsen number in the EC and DC and is taken from [2]. It should be pointed out that two different characteristic lengths are used for each chamber: the nozzle aperture ($L = 0.05\ m$) and the box width ($L = 0.2\ m$). The resulting bands identify a range where Kn is expected to be found. Indeed, as it is stated in [2] and [1], a better expression that can be used for the characteristic length is:

$$L = \frac{\rho}{|\nabla \rho|} \tag{2.2}$$

where ρ is the density. From the point of view of a moving particle, in fact, what is relevant is not the physical dimension of the system, but the macroscopic grandients characterizing the density distributions.However, in a 0D approach only the geometrical lengths of the system are available. The approach employing a band

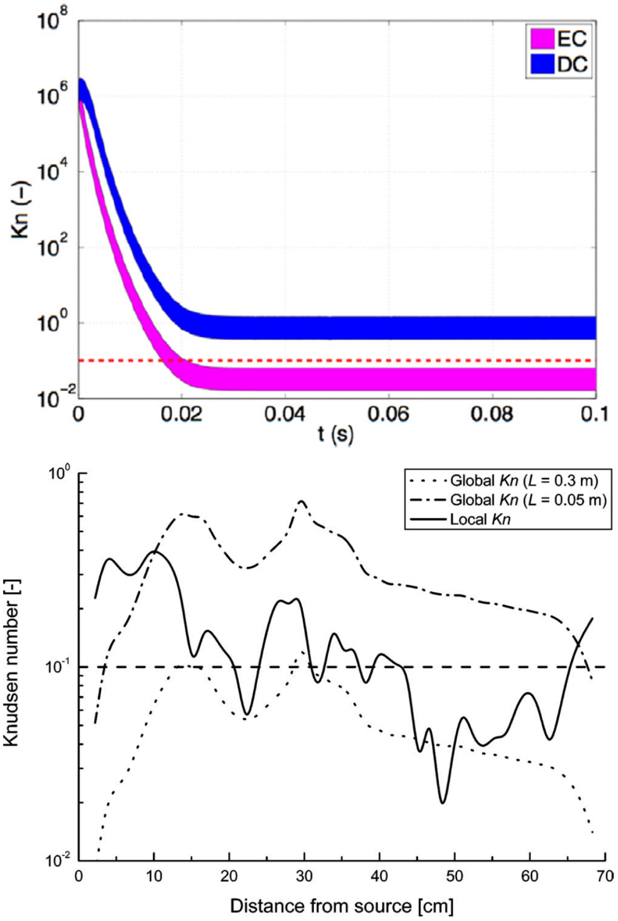

Figure 2.2: On the left, the time evolution of Kn in a 0D approach. On the right, Kn as a function of the distance from the source in a linear plasma generator. Adapted from [2] and [3]

evaluated by using the aperture width and the extension of the chamber as «limiting» characteristic lengths is found to provide a proper bracketing of the actual Kn for differentially pumped system [3] (see Figure 2.2 right).

By looking at the left plot, the expectations discussed above are confirmed: the EC band is below the slip flow regime upper bound (i.e. $Kn = 0.1$) and can be modeled by means of the modified Navier-Stokes equations. Opposite, The DC band is such that a transitional regime is expected. A DSMC approach is therefore required, as conventional CFD codes employing Navier-Stokes equations are not valid.

The right plot of Figure 2.2, which is taken from [3], shows the Knudsen number as a function of the distance from the source in a linear plasma generator. From the conceptual point of view, this device is opposite to a box type divertor. Indeed, a plasma beam is produced by a source and impinge on the target. The aim is that of studying Plasma Surface Interaction (PSI) and this explains the presence of skimmers, which are required to reduce as much as possible the amount of neutrals that can contaminate the beam.

It can be noticed that the local Kn distribution (i.e. that obtained using equation 2.2) is between the curves associated to the nozzle and box radii (i.e. $L = 0.05\ m$ and $L = 0.3\ m$ respectively - the device is axysimmetric), oscillating around the slip regime upper bound. As a consequence, the DSMC approach is a good choice to model the system.

In conclusion, the DSMC method has been already used in literature to perform studies on applications that are close to the box type divertor. As a consequence, this model is adopted for the present analysis and is discussed in the next section.

2.2 DSMC method

The Direct Simulation Monte Carlo technique, developed by Bird (see [4]), is a molecular method that differs from Molecular Dynamics (MD) methods because of the way interactions between particles are modeled. A detailed description of the differences between DSMC and MD is given in [5] and is briefly summarized below.

According to MD, the motion of an arbitrary particle is the result of the combination of the effect of its velocity and of the resulting force acting on it. This is what happens in reality and is described by the laws of classical mechanics. First, the particle moves in space for a small time dt; then, it is performed a check aimed at verifying if intermolecular collisions have happened. If this is the case, the velocity vector is modified so that the interaction is accounted for. At this point, the particle moves again for another time interval and the procedure is repeated. The key feature of the MD technique is that two particles interact with each other if and only if their trajectories intersect (in the present thesis it is assumed that collisions can be described by means of the hard sphere model. This assumption is consistent with the fact that Li vapor is mostly monoatomic [1]).

On the other hand, DSMC uses a statistical approach to account for particles interactions. After the initialization is done, particles are moved for a time dt and their motion is tracked, no matter if, in reality, collisions can happens during this interval. It is as if particles whose trajectories intersect are transparent with each other during the motion. After the new particles positions are identified and indexed, interaction are statistically accounted for.

Each computational cell of the domain will contain a sufficiently large amount of particles (this requirement is discussed in section 2.3); in particular, the probability of particle «*i*»to collide with particle «*j*»in a fixed cell is proportional to the magnitude of their relative velocity. In order to better clarify this statement, let us imagine to have two particles moving at velocities $\vec{u_i}$ and $\vec{u_j}$ having opposite sign. The probability to collide will depend only on their eventual collinearity. Conversely, if they moves in the same directions, the collision probability also depends on the magnitude of the relative velocity $\vec{u_r}$ (e.g. if $|\vec{u_i}| = |\vec{u_j}|$, they can not interact). As a result, it is reasonable to state that the higher $|\vec{u_r}|$, the higher the collision probability.

The previous approach would be computationally expensive, because each particle pair should be evaluated for each cell. As a consequence, Bird suggested a technique named No-Time-Counter (NTC), that is based on acceptance-rejection sampling and is cheaper from the computational point of view. It is not the aim of this work to discuss the NTC, for which specific literature is suggested [1] and [4].

It is relevant to point it out that, although DSMC is computationally cheaper than MD, especially for fluids that are in the transitional regime, the laws of classical mechanics are not all verified. In fact, as it is well discussed in [5], the conservation of angular momentum can no longer be satisfied. This issue is particularly important when dealing with vortical flows, which is not the case in a box type divertor. Therefore, we believe that DSMC is appropriate for describing the situation of a LM vapor box divertor.

Once the particle tracking and the binary collisions are evaluated, the DSMC method perform the sampling of the macroscopic thermodynamic variables (i.e. temperature, pressure, density and velocity). The procedure is based on the use of time-averaged particle properties, as it is explained in [1].

As a matter of fact, two kind or errors are present: those associated to the statistic and the deterministic error caused by the discretization in time and space (see [6]). Statistical analysis errors are evaluated in [7], where expressions for variances and relative standard deviations are computed by employing equilibrium statistical mechanics. These expressions estimating the error in the velocity, density, temperature and pressure are then validated using both DSMC and MD. Unfortunately, due to the limited time available for the thesis, a detailed error analysis has not been performed and will be the object of future work.

Algorithm 2: Direct Simulation Monte Carlo method

```
Particle initialization inside the domain
while $(t \leq t_{end})$ do
    particle tracking
    binary collision
    sample properties
end
Compute averaged properties
```

The algorithm shown below summarizes the structure of the DSMC method that has been discussed.

2.3 Evaluation of the cell dimensions, timestep and scaling factor

Since the DSMC exploits some ideas that are typical of the CFD (i.e. a mesh is required to properly compute binary collisions), it is mandatory to evaluate the grid dimensions, the adopted time step and the Number of Particles per Cell (NPC) which generate proper simulation accuracy. This has been done in [6] and is reported below.

From the kinetic theory it is possible to express the mean free path λ and the particle average velocity $\overline{u}$ as functions of the thermodynamic variables:

$$\lambda = \frac{1}{\sqrt{2}\pi d^2\, n} \tag{2.3}$$

$$\overline{u} = \sqrt{\frac{3\, k_B T}{m}} \tag{2.4}$$

where d is the particle diameter and πd^2 the collisional cross section (in equation 2.3, the factor 1/4 is replaced by $\sqrt{2}$ because a particle can be both projectile and target), T is the temperature, k_B the Boltzmann constant, m the particle mass, n the particle density.

The grid size should always be less than λ, otherwise different collisions associated to the same particle can happen inside the same cell and this leads to a bad accuracy. Similarly, the time step should be less than $\lambda/\overline{u}$, in order to capture collisions that happen in short times. In [6] the following recommendations are suggested:

$$\Delta x \leq \lambda/3 \tag{2.5}$$

$$\Delta t \leq \lambda/\left(10\,\overline{u}\right) \tag{2.6}$$

and are used within this work. Using equations 2.5 and 2.6, in fact, the Courant number is equal to 3/10, thus respecting the CFL condition. In other words, the time interval needed for a particle to travel across an arbitrary computational cell is higher than the chosen timestep. This implies that the numerical resolution is accurate

The scaling factor f is another important parameter that has to be introduced. It is defined as the amount of real particles that each DSMC particle represents and is evaluated as:

$$f = \frac{(\Delta x)^2\ n}{NPC} \tag{2.7}$$

f can not be arbitrarily large. Indeed, if this had been the case, a user could have decided to set it equal to the whole real population that should be simulating, resulting in a single DSMC particle. In such a situation, the relative standard deviations associated to the thermodynamic variables would have been too large and results would not have been relevant. As a consequence, the scaling factor has an upper bound; in [4] there are some examples using a value $\sim 10^{12}$, which can be seen as a rough estimation of the upper limit f_{lim}.

If, after having set the grid dimensions and the timestep according to the foregoing reccomendations, it occurs that $f \geq f_{lim}$, NPC, which can be seen as a free parameter, is increased in order to reach the condition $f < f_{lim}$. NPC has in fact has a lower bound (that is assumed to be in the range 5-7 — see [4] and [6] for details), but has not an upper bound. Indeed, the higher NCP, the closer the amount of DSMC particles to the real ones. Nevertheless, the higher the computational cost. It is thereby assumed that NPC≤ 50. Whenever this limit is exceeded the mesh size Δx is decreased to get to reach the condition $f < f_{lim}$. However, such a procedure leads to an increase of the number of cells and this also means that the computational cost increases. Nevertheless, the effect is smaller than that obtained further increasing NCP.

At this point, some considerations should be done:

1. the mesh size of a generic cell that is in the EC is much finer than that of a cell of the DC, because the particle density is much lower there;
2. none of the thermodynamic variables is known *a priori.* Thus, equations 2.5, 2.6 and 2.7 can't be directly exploited. In order to solve the problem, the values resulting from the 0D model are used. It will then be verified *a posteriori* if this choice is conservative or not. This allows a considerable saving of time during the simulation setup phase.

2.4 Conclusions

The Knudsen numbers characterizing the EC and the DC of a box type divertor lead to the conclusion that the fluid can be seen almost as a continuum in the EC, not in the DC, since it is expected to be in a transitional regime and therefore molecular simulation methods should be used.

Among the different techniques, Direct Simulation Monte Carlo (DSMC) is chosen. The main difference between DSMC and more common Molecular Dynamics (MD) methods is that collisions are described in a probabilistic manner and a use of a grid is required to mesh the computational domain. Inter particle interactions are not evaluated during the particle tracking, but through the acceptance-rejection NCT technique that has been developed in [4]. This approach is much more computationally effective than MD, especially in a system as the one which is being under study, where it exists a region of relatively high density which would be prohibitive to model with MD. An analysis of the statistical errors is mandatory, but it is not done in this work because of the limited time available. The procedure to be used is that described in [7].

The mesh size, timestep and scaling factor are finally evaluated using the values resulting from the 0D model, which will be verified *a posteriori.* In particular, the significant density gradient existing between the EC and DC leads to a choice of the mesh size that is much coarser in the latter chamber, thereby reducing the computational cost of a simulation.

References

[1] C. White et al. «dsmcFoam+: An OpenFOAM based direct simulation Monte Carlo solver». In: *Computer Physics Communications* (2017).

[2] G.F. Nallo et al. «Modeling the lithium loop in a liquid metal pool-type divertor». In: *Fusion Engineering and Design* (2017).

[3] H.J.N. van Eck et al. «Modeling andexperiments on differential pumping in linear plasma generators operating at high gas flows». In: *Journal of applied physics* (2009).

[4] G.A. Bird. «Molecular gas dynamics and direct simulation of gas flows». 1994.

[5] E. Meilburg. «Comparison of the molecular dynamics method and the direct simulation Monte Carlo technique for flows around simple geometries». In: *Physics of fluids* (1986).

[6] Z.X. Sun et al. «Proper cell dimension and number of particles per cell for DSMC». In: *Computers and fluids* (2011).

[7] N.G. Hadjiconstantinou. «Statistical error in particle simulations of hydrodynamic phenomena». In: *Journal of computational physics* (2003).

Chapter 3

DSMC simulations

The numerical code adopted to perform the simulations is called *dsmcFoamPlus* and has been developed from «dsmcFoam», a well established solver implemented in OpenFOAM since 2010 [1]. The major differences regard the implementation of a large amount of new BCs and macroscopic property measurements tools. dsmcFoamPlus also has better performance than dsmcFoam in terms of parallel processing.

It is not within the scope of the present thesis to discuss the implementation of the code, for which specific references are suggested [1] and [2]. Indeed, the solver has been verified with analitycal expressions in [1] and is therefore assumed to be correct.

The assumptions on which numerical simulation are based are first presented. After that, the application to a LM divertor is discussed. In particular, two different cases are evaluated, one having diffusely reflecting separating surfaces near the nozzles, the other having condensing separating surfaces there. The setup is analyzed and BCs are explained one by one; after that, results are shown.

An optimization procedure is finally performed to establish how the shape of the nozzles influences the distributions of pressure, temperature and density inside the chambers.

3.1 Assumptions

As stated in section 2.1, Knudsen numbers computed through the 0D model are such that a transitional flow regime is obtained. This will be verified *a posteriori* and justifies the choice of DSMC, whose main assumptions are listed below. Actually, they have already been mentioned in the previous chapters, but they are repeated for the sake of clarity.

1. *monoatomic fluid*: the working fluid is lithium, that is assumed to be monoatomic.

As a consequence, the isoentropic coefficient γ is $5/3$. Vibrational and rotational contributions can be neglected, so translation energy is the only relevant quantity for the analysis;

2. *condensing* and/or *evaporating mass fluxes*: to model these physical phenomena, a specific BC (described in the next section) is used. This BC is strictly related to the Hertz-Knudsen formula (that is its integration over the domain);

3. the timestep and mesh sizes of each chamber are chosen based on the expression presented in the previous chapters and computed using the results produced by the 0D model. This choice will be verified *a posteriori*;

4. pressure, temperature and density BCs are of Dirichlet type. Their numerical values are consistent with the ones employed in the 0D model and are also used to initialize the simulations.

Finally, it should also be mentioned that we are interested in the steady state, so the results (shown in the next section) refer to this situation.

3.2 Box type divertor

This section deals with the simulation of the vapor metal inside the box type divertor, that is the original aim of the present work. The geometry is produced in the same way of that described in chapter 5, although the mesh is no more trimmed (cubic cells), but triangular (tetrahedral). Moreover, because of the expected densities in the two chambers are different by order of magnitudes, the number of cells in the EC is much larger than in the DC.

The working fluid is lithium, while the BCs are different because two different cases are analyzed. A detail discussion of the simulation setup is therefore the object of the next paragraph.

3.2.1 Simulation setup

Figure 3.1 shows the mesh and summarizes the simulation setup. A zoom is performed around the nozzle between the EC and the DC in order to show that the mesh in the latter chamber is much coarser than that in the first chamber.

As already anticipated above, two different cases are studied and then compared each other:

1. condensing separating surfaces;

2. diffusely reflecting separating surfaces.

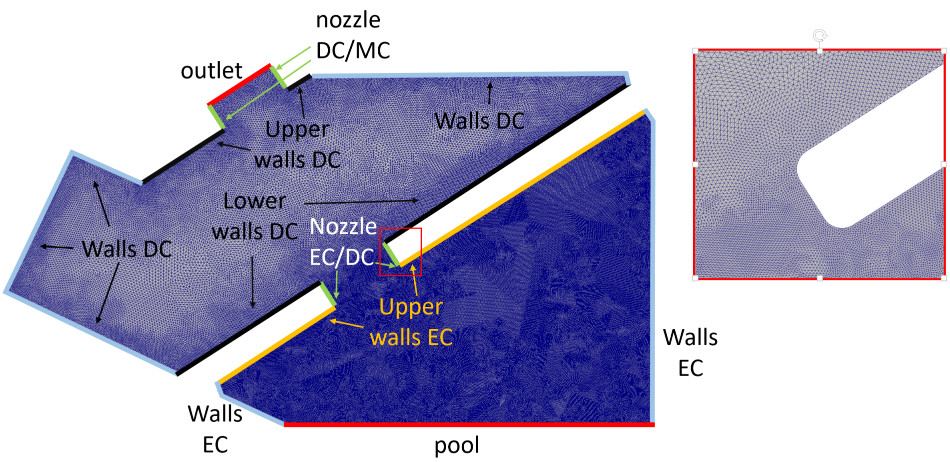

Figure 3.1: Mesh with zoom of the nozzle between EC and DC

Therefore, for the sake of clarity, BCs are classified according to the implementation of the code and then described one by one.

- *dsmcPatchBoundaries*: this kind of BCs is characterized by the fact that particles are followed from the inside towards the outside (i.e. the boundaries). As a result, it can be either absorbed or, at most, reflected. No injection is provided. Two different dsmcPatchBoundaries are used here:
  - *dsmcDiffuseWallPatch*: it is a diffuse reflection (not a specular reflection). A particle impinging on a boundary having such a BC is thermalized and then reflected with a velocity $\vec{u} \overset{\text{def}}{=} (u_x\, u_y\, u_z) \overset{\text{def}}{=} (\phi\, \theta\, |\vec{u}|)$ that is such that:
    - ∗ the direction is given in terms of the poloidal and azimutal angles θ and ϕ according to an isotropic emission;
    - ∗ the magnitude is given according to a maxwellian distribution whose mean value is the wall temperature T_w. It should be remembered, in fact, that in the non-relativistic limit (as in the present case) it is $|\vec{u}| = \sqrt{k_B\, T_w}/\,(2\,m)$, where k_B is the Boltzmann constant and m the particle mass, that is a known quantity (the fluid type is fixed).
  - *dsmcDeletionPatch*: it is a pure absorption.
- *dsmcGeneralBoundaries*: this type of BCs is responsible for the intake of particles, not caring of any motion. Only one BC belonging to this group is used here:
  - *dsmcFreeStreamInflowPatch*: the input required are:
    - ∗ translation temperature: since particles are assumed to be mmonoatomic, vibrational and rotational energies are not relevant. As a consequence, fixing this quantity implies to fix the magnitude of the velocity the

particle has with respect to the boundary (i.e. it is the relative, not absolute, velocity);

- velocity: it is the boundary velocity. Since in the present case there are only walls that can emit, this velocity is set up to zero. Therefore, the absolute velocity of a particle is completely determined by the translation temperature (the emission is assumed to be isotropic);
- density: this quantity gives the amount of particles that are emitted. To be more precise, temperature and pressure at the walls are fixed and equal to those at saturation; the density at the boundary is fixed so that a sufficient number of particles is emitted from the wall to maintain the desired pressure. The dsmcFreeStreamInflowPatch BC provides thereby an amount of particles that always guarantees that constant density at the walls (its value is given by the user).

In order to simulate condensation/evaporation walls the FreeStream and Deletion BCs are applied together (from the point of view of the code, these input commands are not mutually exclusive). If the amount of absorption is higher than the correspondent emission, then the surface is evaporating; otherwise, it is condensing. Thus, such a combination role is exactly equivalent to the Hertz-Knudsen formula.

From the engineering point of view, dsmcDeletionPatch corresponds to non condensing walls (no cooling is provided at those surfaces having such BC), while the combination dsmcFreeStreamInflowPatch/dsmcDeletionPatch (from now on it will be called *evaporating/condensing*, depending on the situation) refers to the opposite situation. Once more, it is fundamental to compare the diffusely reflecting and condensing where simulations, because they reflect different engineering designs (i.e. establish the amount of cooling tubes needed).

Table 3.1: Boundary conditions for each surface. On the left, the condensing simulation. On the right, the diffuse reflecting one.

Boundary	Type of BC	T_w [K]	p [Pa]
pool	evaporating	950	$p_{sat}\,(T_w)$
walls EC	condensing	519	$p_{sat}\,(T_w)$
walls DC	condensing	466	$p_{sat}\,(T_w)$
nozzle EC/DC	reflecting	550	-
upper walls EC	condensing	519	$p_{sat}\,(T_w)$
lower walls DC	condensing	466	$p_{sat}\,(T_w)$
nozzle DC/MC	reflecting	500	-
upper walls DC	condensing	466	$p_{sat}\,(T_w)$
outlet	pure absorber	-	-

Boundary	Type of BC	T [K]	p [Pa]
pool	evaporating	950	$p_{sat}\,(T_w)$
walls EC	condensing	519	$p_{sat}\,(T_w)$
walls DC	condensing	466	$p_{sat}\,(T_w)$
nozzle EC/DC	reflecting	550	-
upper walls EC	reflecting	550	-
lower walls DC	reflecting	500	-
nozzle DC/MC	reflecting	500	-
upper walls DC	reflecting	500	-
outlet	pure absorber	-	-

Table 3.1 summarizes the BCs for each surface of the divertor.

3.2.2 Results

Figure 3.2 shows the density distribution, expressed as [particles/m^3], of both the condensing and the diffusely reflecting simulations. It can be noticed that the amount of particles inside the system reflecting case are much more than those in the condensing one. This is due to the presence of a lower number of sinks for the vapor; indeed, the upper surfaces of the EC do not absorb anything, as it happens for both the upper and lower surfaces of the DC. As a consequence, the mass flow rate through the outlet is larger than that of the first case. Doing the post processing of the results it has been found that:

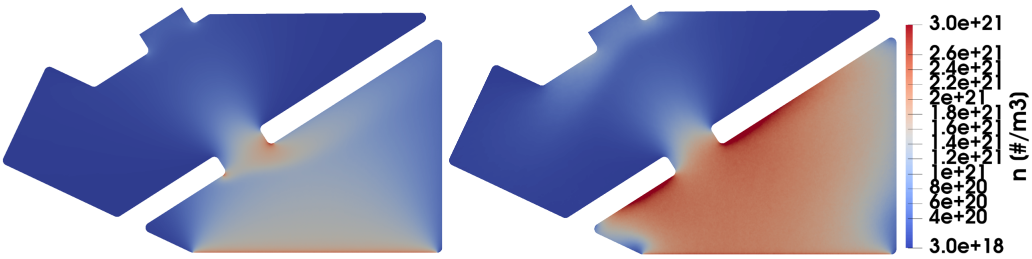

Figure 3.2: Density distribution inside the divertor. On the left, the condensing simulation. On the right, the diffuse reflecting one.

$$\dot{m}_{refl}^{out} = 1.3\,\dot{m}_{cond}^{out} \tag{3.1}$$

where the subscripts $refl$ and $cond$ refers to the specific simulation.

The previous result would lead to conclude that the condensing setup is better than the reflecting one. However, as it can be seen by looking at Figure 3.2, the latter case confines much better the metal vapor. Thus, radiation power losses (discussed in chapter 1) are much more relevant. This would potentially lead to a temperature reduction inside the EC so high that the evaporating mass flux from the pool might significantly decrease. To reach a conclusion, this modeling should also foresee a feedback from the plasma, which is however beyond the scope of the present thesis.

A relevant feature which is visible when looking at Figure 3.3 is that pressure at upper walls of the DC is significantly higher in the reflecting simulation. This is in agreement with the theory of normal shock waves (see [3] for details) and confirms what has been already anticipated in section 1.1.1.

Figure 3.4 shows the velocity magnitude for the two simulations. The first thing to notice is that the condensing case has a more uniform distribution. This is a direct consequence of the different BCs; in fact, in the diffusely reflecting case that are less vapor sinks, so the particles could reach a stagnant flow regime in an easier way.

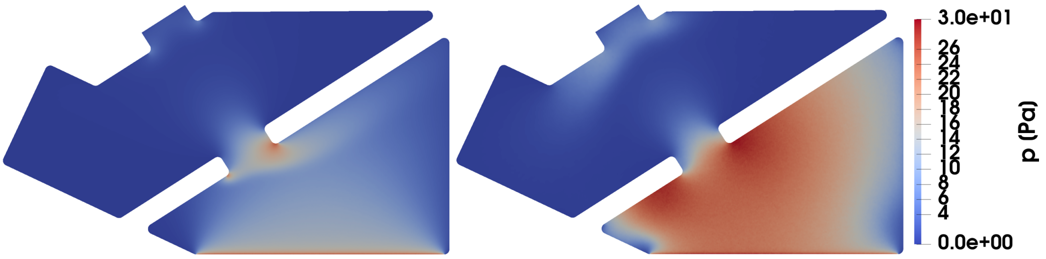

Figure 3.3: Pressure distribution inside the divertor. On the left, the condensing simulation. On the right, the diffuse reflecting one.

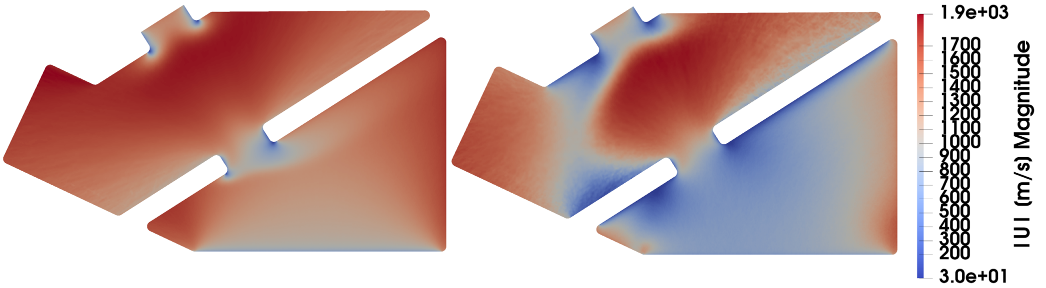

Figure 3.4: Velocity magnitude distribution inside the divertor. On the left, the condensing simulation. On the right, the diffuse reflecting one.

However, the most relevant feature is the huge gradient that is in front of the nozzle DC/MC. This confirms once more the presence of a supersonic shock Velocity decreases so sharply because the upper surfaces of the DC reflect particles back; this compresses the vapor flow moving towards the outlet, that is expanding. When passing from the EC to the DC, in fact, particles feel a huge pressure gradient and they increase their velocity as far as they do not arrive close to the nozzle DC/MC, where they are reflected. The shock arising from the consequent compression has a well defined shape, as it can be seen by looking at Figure 3.4, right.

Figure 3.5 shows the Mach number and the streamlines (black lines). It is interesting to notice how different the particles trajectories are near the reflecting walls. Furthermore, it can be noticed that in the DC there is a highly supersonic flow, especially in the condensing case. This comes from the fact that very few particles reach the corners of the DC, so pressure and temperature have also low numerical values (we are closer to vacuum situation). It can also be stressed that statistical errors are expected to be higher at the DC corners, since there is a lower amount of information available.

Figure 3.6 shows the temperature distribution, which agrees with all the comments that have been done for the other thermodynamic variables. In the reflecting case T is significantly larger than in the condensing one; this directly comes from

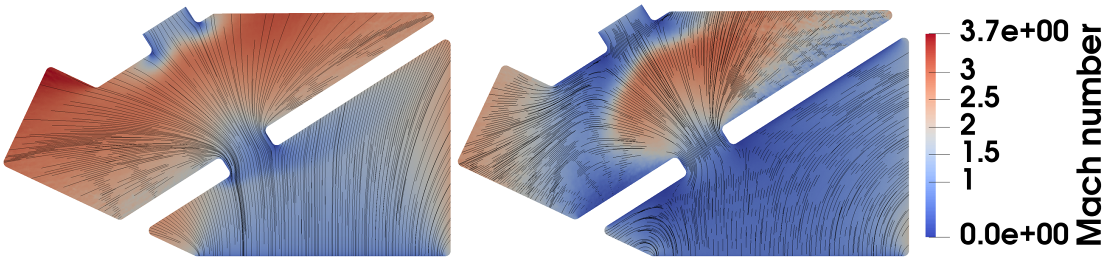

Figure 3.5: Mach distribution and streamlines inside the divertor. On the left, the condensing simulation. On the right, the diffuse reflecting one.

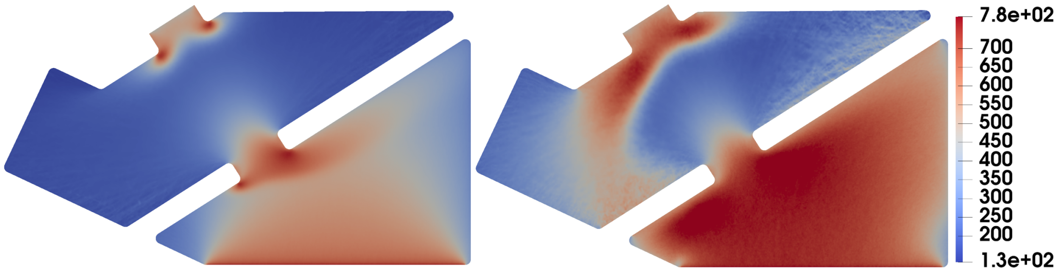

Figure 3.6: Temperature distribution inside the divertor. On the left, the condensing simulation. On the right, the diffuse reflecting one.

the features already discussed for density and pressure. The same happens after the wave front it increases, as it is expected from normal shock theory [3].

3.2.3 Comparing with the 0D results

The right plot of Figure 3.7 shows the particle density along the probe line that is shown on the left. The first thing to notice is that the density inside the EC in the reflecting case is almost two times that of the 0D model. This means that the mean free path computed using equation 2.3 is half of that used to evaluate the grid size. However, thanks to the recommendation suggested in equation 2.5, it can be concluded that the mesh is adequately fine to catch the particle motion between two successive collisions. Therefore, no further iterations on mesh generation procedure are performed.

The same conclusion can't be extended to the DC. By looking at Figure 3.7 it is in fact clear that the density computed in both the 2D simulations is much larger than that of the 0D (more that a factor 2). However, it should be noticed that this feature refers to a specific probe line, which passes through the nozzles. As a consequence, it might be possible that globally the mesh is sufficiently fine in the DC. However, it can be stated that the «channel» going from one nozzle to the other inherit the characteristic grid size of the EC rather than that of the DC.

Furthermore, it seems relevant to notice that the 0D model is conservative: a

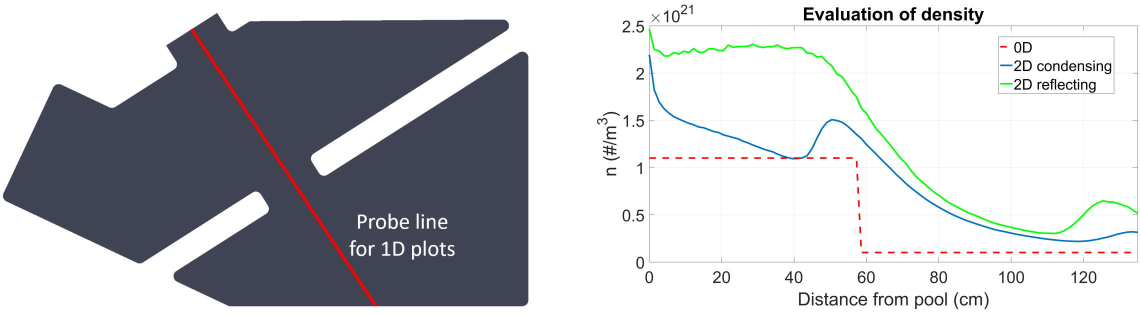

Figure 3.7: On the left, the geometry with the probe line used for 1D plots. On the right, the density along the probe line.

lower density inside the EC implies that less particles interact with the incoming plasma, so radiation losses are weaker and this causes a localized thermal load on the pool which is higher.

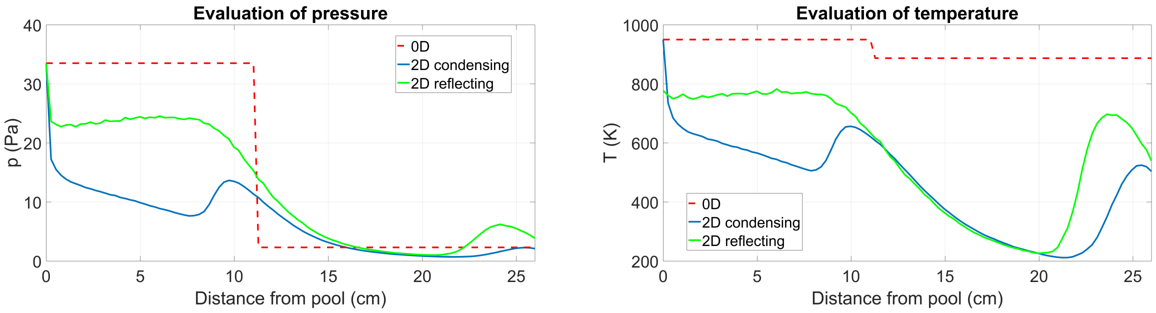

Figure 3.8: Pressure and temperature distributions along the probe line.

Figure 3.8 shows the pressure (left plot) and the temperature (right plot) distributions along the probe line. As it has been discussed in 1.1.1, inside the EC both the thermodynamic variables are overestimated. This is caused by:

- the supersonic expansion: looking at the right plot, it can be seen that between $10\ cm$ and $20\ cm$ (distance from the pool) there is a constant reduction of temperature. This comes from the supersonic expansion the fluid undergoes passing from the EC (where pressure is high) to the DC (where pressure is low);
- the 0D model assumes that $p = p_{sat}$ and $T = T_{sat}$, where sat represents saturation conditions at the pool, in the EC, that is not true due to the presence of condensing walls.

Another important feature visible looking at the right plot is the effect of the shock: there is a sharp increase of T that is more significant in the reflecting simulation. This is in agreement with the shock theory [3].

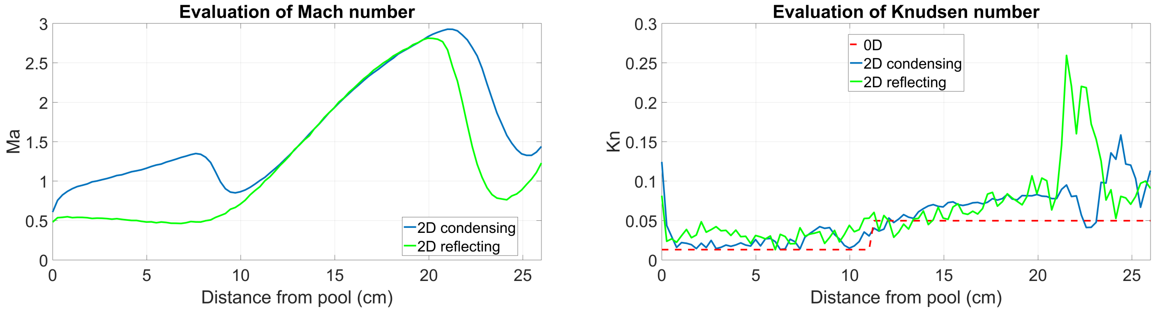

Figure 3.9: Mach and Knudsen and numbers distributions along the probe line.

The left plot of Figure 3.9 shows the Mach number distribution. A first increase associated to the fluid expansion inside the EC is followed by a first shock, that is located at the outlet of the nozzle EC/DC (see Figure 3.5). After that, there is another increase in the DC, which is sharper than the previous one because that chamber is closer to a vacuum condition, meaning that:

- the sound speed is lower since temperature is lower (see Figures 3.6 and 3.8);
- the fluid is expanding, so pressure energy is converted into kinetic energy.

Finally, a second shock, much more significant than the first one, happens at the inlet of the nozzle DC/MC. In particular, the jump that is in the diffusely reflecting simulation is sharper than that of the condensing case.

The right plot of Figure 3.9 shows instead the Knudsen number. The 0D model underestimates the results of both the reflecting and the condensing 2D simulations. This means that the 0D model is conservative in terms of highlighting the necessity to have a molecular approach to face the problem.

3.3 Optimization of the shape of the nozzles

It has been verified that in a box type divertor the fluid reaches a supersonic regime, leading to a potential shock that is larger in the diffusely reflecting configuration than the condensing one. Because of this, an optimization process is performed to correctly design the nozzles. A proper choice would localize the wave front at the beginning of the DC, thus increasing the pressure difference existing between the outlet and the EC, thereby promoting the differential pumping. Before that, it is required to briefly review the theory of nozzles, whose main reference is [3].

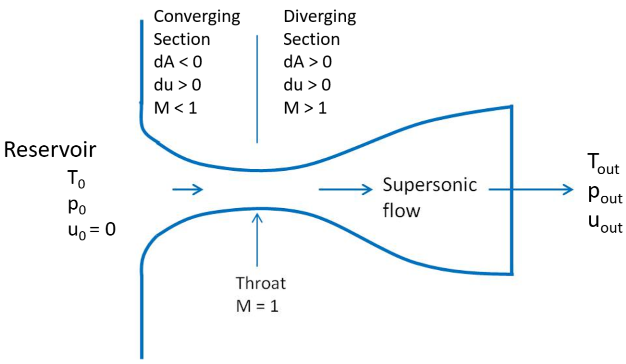

Figure 3.10: Schematic view of a supersonic nozzle. Adapted from Wikipedia.

3.3.1 Shock theory in a nutshell

Figure 3.10 shows a schematic view of a supersonic nozzle. The fluid flows from an ideal infinite reservoir, where it is $p = p_0$ and $|\vec{u}| = 0$, to an external environment at $p_{out} < p_0$. It can be shown that a convergent nozzle is required to increase the fluid velocity in a subsonic regime; however, when sonic conditions are reached, a diverging shape is needed in order to keep the expansion on. Pressure energy is converted into kinetic energy (the flow is assumed to be isenthalpic, since no heat sources/sinks are present).

If the Mach number were less than one at the throat, the following part of the nozzle would behave as a diffuser, thus decreasing the fluid velocity (opposite situation than before). Conversely, if $Ma = 1$ at the throat, the expansion can go on and velocity keeps increasing, leading to a supersonic regime. At this point, two different situation can happen:

1. p_{out} is that associated to the isoentropic condition, which can be computed analytically (see [3]). Th expansion goes on and velocity is increased as it happens between the reservoir and the throat;

2. p_{out} is different from that associated to the isoentropic condition. In this case, it is not possible to get any isoentropic solution. There is a normal shock that localizes in a position in such a way that the outlet fluid pressure is p_{out}. The higher p_{out}, the closer to the throat the wave front, in order to give the fluid more time to compress again.

In a box type divertor the second condition is the one which must be considered, since it is $p_{out} \sim 0\ Pa$. Shocks are therefore localized close to the outlet, as it is clear by looking the Figures reported in the previous section. In this context, the optimization procedure wants to study how the shape of the nozzles influence the positioning of the wave front.

3.3.2 Simulations with different nozzle shapes

Figure 3.11 shows three meshes associated to the following configurations:

- converging nozzles with an angle of 5° with respect to the axis;
- diverging nozzles with an angle of 20° with respect to the axis;
- «neutral» nozzles, i.e parallel to the axis.

each being zoomed at the nozzle ED/DC. In particular, the designing criterion of the apertures is that the smallest cross section is constant. It has been explained in the introduction that the aperture width is fixed by a trade-off in this kind of system. This notion is retained when comparing different nozzle shapes, not withstanding the simplified geometry. In other words, the nozzle outlet in the converging setup and the nozzle inlet in the diverging are both equal to the cross section of the neutral case.

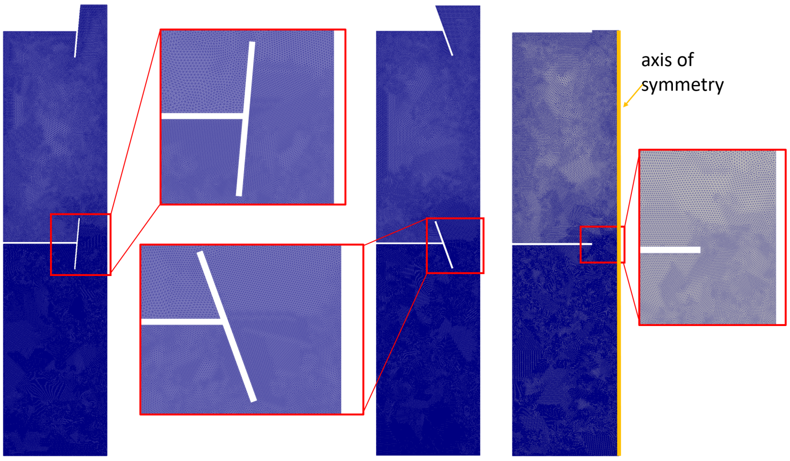

Figure 3.11: Meshes for convergent, divergent and neutral nozzles.

It can be noticed that the geometry is different from that of section 3.2. Indeed, the goal of the analysis does not depend on the shape of the chambers, so the results that will be obtained can be considered general.

The simulation setup is equal to that of the box divertor, with the exception of the right boundary, that is the axis of symmetry of the domain. A symmetry BC is employed there in order to halve the computational effort. As before, two kinds of simulations are performed: condensing and diffusely reflecting.

Results of the condensing case

Figure 3.12 shows the streamlines and the Mach distributions in both the domain and along the axis of symmetry, together with density. By looking at the left plots, it

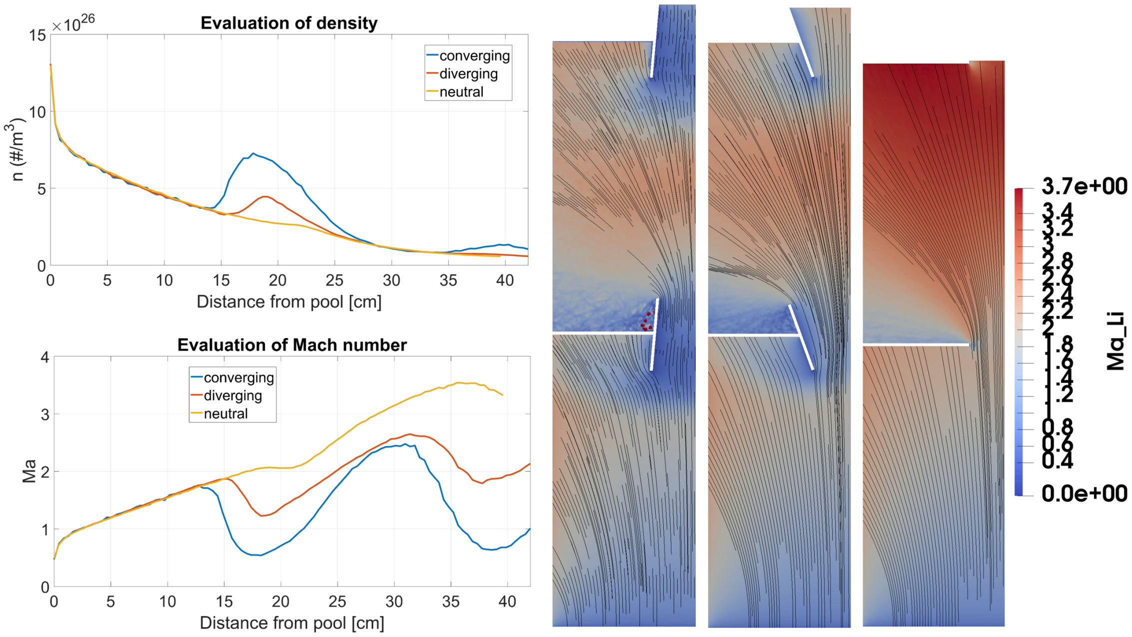

Figure 3.12: On the left, pressure and Mach number along the axis of symmetry. On the right, streamlines and Mach distributions in the domain.

can be noticed that both pressure and Mach grandients reach the highest values in the converging case; this can be understood looking at the streamlines (right part of Figure 3.12). While the neutral and diverging setups do not constrain the fluid exiting from the EC/DC nozzle, in the last case streamlines are compressed as they move closer to the DC. The expansion is limited to the upper part of the chamber and this results in larger variations of density, pressure, and temperature (the last ones are not shown).

Another feature that can be seen from the left plots is that Mach and density gradients are closer to the pool for the converging setup, while they are further localized in the neutral case. This situation reflects the presence of diffusely reflecting surfaces at the nozzles (only the EC and DC upper walls and the bottom DC walls are condensing). As before, the effect is more significant in the converging than the diverging setup because, due to the fluid compression, a particle moving from the EC towards the DC sees a larger reflecting area. On the other hand, when comparing the converging and neutral situations there is a difference which arises from the lack of a sufficiently large surface in the neutral geometry.

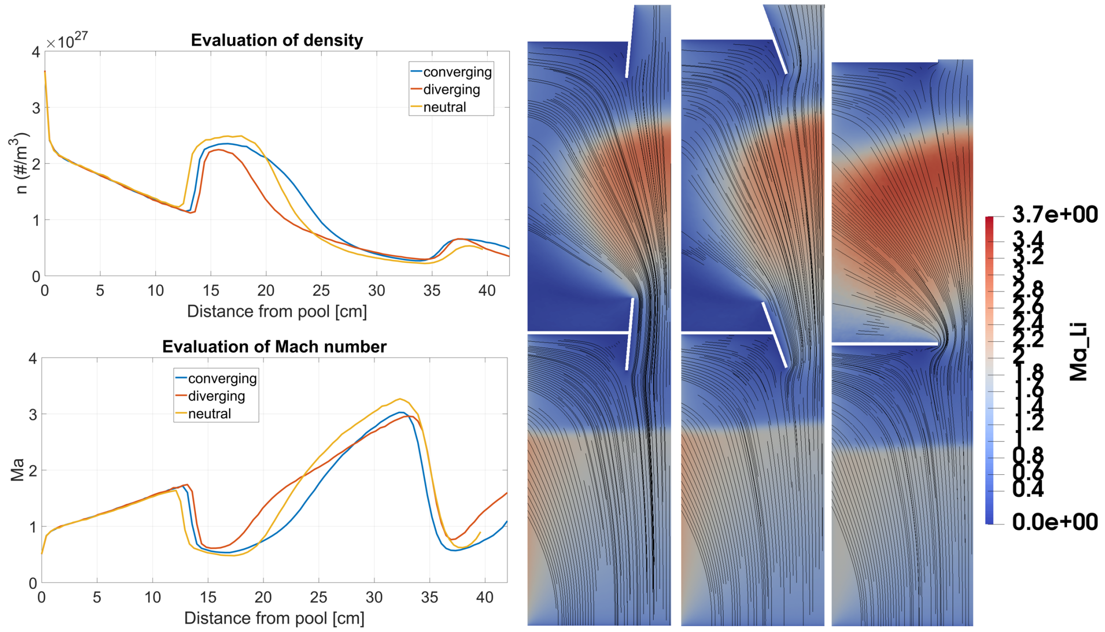

Figure 3.13: On the left, pressure and Mach number along the axis of symmetry. On the right, streamlines and Mach distributions in the domain.

Results of the diffusely reflecting case

Figure 3.13 is like the previous one, but refers to the diffusely reflecting simulations. The same comments as before can be made, although some further considerations must be done. When passing through the nozzles, thermodynamic variables change by a factor of 2 between the EC/DC, by a factor of 3 between the DC/MC. These order of magnitudes are similar to those of the condensing simulations. However, in the reflecting simulations:

- the positioning of the shocks is identified more easily, because variations are sharper;
- absolute values of gradients are twice those of the condensing setups.

It should in fact be remembered that the diffusely reflecting simulations globally have a larger amount of particles (10^{27} vs. 10^{26} particles/m^3), because the amount of sinks available for the vapor to condense is lower. As a consequence, variations are larger. This feature might be helpful when dealing with shocks; in particular, if the positioning of the first wave front is sufficiently closer to the EC, the sharp increase of density may result in higher radiation losses, thus redistributing the thermal loads impinging on the pool over a larger area.

Conclusions

As far as diffusely reflecting simulations are concerned, the shape of the nozzles does not influence the thermodynamic variables distributions inside the LM divertor.

Opposite, the design of these components may affect the results in a condensing case. In fact, this situation is characterized by a lower amount of particles inside the system, so the effect of compression caused by a converging nozzle might help in localizing the shocks closer to the EC. Moreover, pressure, temperature and density variations are more significant and this might be relevant in redistributing the thermal loads impinging on the pool on a wider area.

3.4 Conclusions

In this chapter, the analysis of a box type divertor has been addressed.

The main assumptions and the simulation setup are firstly discussed, giving particular emphasis to the described of the boundary conditions. Two different cases are presented and compared to each other: condensing and diffusely reflecting. In particular, the latter globally has an amount of particles that is one order of magnitude larger than the other case. This leads to a mass flux through the EC/DC nozzle that is 1.3 times the other one. However, the higher density inside the EC enhance the radiation losses, thus reducing the local peak load impinging on the liquid pool.

2D results are then compared with those coming from the 0D model and the Knudsen number, the grid size and the timestep are verified *a posteriori*. In particular, the choice of Δx ins the EC is verified to be conservative, while in the DC a finer mesh is required, because of the high density in the «channel»that is between the two nozzles.

The 0D model underestimates the particle density both in the EC and the DC. This is conservative because a lower amount of particles implies a smaller effect of radiation losses in the EC and, as a consequence, an higher peak load.

On the other hand, pressure is overestimated in the 0D model, which does not account for shocks.

Indeed, it is found that the fluid can reach a supersonic regime, undergoing a shock. An optimization procedure is therefore suggested for both the condensing and the diffusely reflecting simulations. After the analysis, it has been found that the shape of the nozzle does not influence the distributions of thermodynamic variables in the reflecting case, while fro the condensing converging nozzles are preferred, since they increase the effect of shocks, whose wave front is localized closer to the pool.

References

[1] C. White et al. «dsmcFoam+: An OpenFOAM based direct simulation Monte Carlo solver». In: *Computer Physics Communications* (2017).

[2] G.A. Bird. «Molecular gas dynamics and direct simulation of gas flows». 1994.

[3] P. Cinella. «Flussi supersonici». 2005.

Part III

Final remarks

Chapter 1

General conclusions and future developments

The purpose of this thesis, which has been divided in two parts, was to perform a 2D analysis of a box pool type divertor, aiming to provide a better evaluation of liquid metal evaporation/boiling and condensation, as well as differential pumping of metal vapor, thereby improving the predictions of the currently available model.

In the first part, a Computational Fluid Dynamic (CFD) model was presented. The main goal was to develop a code which properly evaluates the phenomena associated to thermal phase change, which have a fundamental role at both the condensing walls and the interface between liquid and vapor phases. In the evaporation chamber (EC), in fact, the fluid is expected to be in a regime where the continuum assumption holds, and therefore CFD is an adequate tool.

The numerical code is developed within the OpenFOAM environment and is based on the Volume Of Fluid (VOF) approach. Therefore, one set of conservation equations (mass, momentum, energy) is written with respect to the mixture (i.e. one fluid). As a consequence, thermodynamic properties are average quantities between liquid and vapor. A fourth equation referring to the volume fraction is added to catch the interface.

However, during the verification procedure it has been found that mass is not conserved. After many analyses, it has been found that the numerical algorithm that is used by the software to reconstruct the interface (i.e. MULES) is not applicable to problems dealing with phase change. Indeed, the reconstructing algorithm is based on the assumption that fluid velocities are divergence free; this is not true at those cells where phase change occurs, because the divergence of the fluid velocity is proportional to the volumetric mass transfer rate. As a consequence, no matter what kind of phase change is under study (e.g. cavitation, evaporation/boiling, condensation), the computational cells where the interface is located are badly solved. Since the interface moves in space as time goes on, all the domain may be affected by this issue, thereby not conserving the mass.

Many other different algorithms (e.g. Piesewise Linear Interface Capturing - PLIC - technique) have been proposed in literature. However, at the time this thesis has been written, none of them were found to properly account for phase change at the interface. As a results, further studies must be done to realize a robust algorithm. This will be the object of future developments.

In the second part, a Direct Simulation Monte Carlo analysis has been performed to investigate the metal vapor dynamics and differential pumping. The DC was in fact found to be in a transitional regime, so a molecular approach is needed to describe the system there.

Two different simulations were discussed, one having all condensing walls, the other with diffusely reflecting walls close to the nozzles. It was found that the latter better confines the vapor metal inside the EC, thereby enhancing the radiation losses and the resulting redistribution of the thermal load over a wider area of the liquid pool. This goes in the direction of reducing the metal evaporation/boiling rate. However, the particle flux passing through the nozzle between EC and DC is 1.3 times larger than that computed in the first case. This comes from the presence of an larger amount of sinks (i.e. condensing walls) and may result in a metal particle flux towards the core plasma that is not sustainable. Therefore, further investigations must be done in order to understand which configuration is the best, i.e. by applying the 2D model in the framework of the self-consistent model, outlined in section 1.1.1.

In both cases shocks were found to be present at the nozzles. Their effect can be fundamental, because a proper positioning of the wave fronts can enhance the pressure difference existing bewteen the EC and the core plasma, thus facilitating differential pumping. Moreover, the density increase that is present after the shocks have place can further help in redistributing the thermal loads thanks to radiation losses. Therefore, an optimization procedure for the nozzles shape was done. It was found that a converging shape would be the best choice in the condensing simulation, while no relevant effects were found in the diffusely reflecting one.

Finally, the 2D results were compared with those coming from the 0D model that was preliminary described. This model underestimates the density inside both the EC and the DC. This goes in the direction of being conservative, since lower densities imply less radiative losses and, consequently, higher thermal peaks. Similarly, temperature is overestimated, as well as pressure in the EC. This comes from the fact that the 0D model assumes thermodynamic equilibrium everywhere in the EC (this hypothesis is only true near the pool) and neglects the effects caused by the aforementioned shocks.

Chapter 2

Ringraziamenti

Dovrei scrivere i ringraziamenti in inglese, essendo la tesi in inglese. Dovrei, allo stesso modo, mantenere un certo decoro, essendo questo documento una tesi di laurea magistrale, per l'appunto. Mi si permetta, tuttavia, almeno a questo punto, di prendermi la libertà di far conoscere un pochino la mia persona all'assiduo lettore che sia giunto fin qui nella lettura. Onestamente, credo che solo l'encomiabile, unico, integerrimo Giuseppe sia arrivato qui partendo dall'inizio. Tu, ipotetico lettore, se non sei Giuseppe (o la prof. Savoldi, o l'ipotetico controrelatore), avrai senz'altro balzato il mio papiro dalla prima all'ultima pagina, e ti sarai ritrovato qui per caso. E' per questo che ti esorto, lettore! Preghiamo insieme e diciamo: «Ascoltaci, o Giuseppe!» Come avrai senz'altro capito, questi ringraziamenti alterneranno momenti di serietà a momenti più scherzosi, nello spirito che caratterizza la mia persona. Come avrai senz'altro capito, non posso non cominciare con ringraziare lui, Giuseppe.

- A Giuseppe, mio supervisore, compagno di viaggio, colonna e sostegno nei momenti di difficoltà, stella polare durante la notte, mentore di Monte Carlo. Alla splendida persona che, umanamente e professionalmente, hai costruito con sudore e sacrifici. Alle discussioni avute, ai confronti, agli «#automatizzatistoc***o» . Dalla «revisione_ tesi_ fino_ a_ na_ certa» alla «revisione_ tesi_ da_ na_ certa_ alla_ fine». Senza di te, che per me sei co-autore a tutti gli effetti, non avrei mai potuto partorire quella che, citando il nostro Enrico Fermi, è definibile come «una delle porcherie più solenni» ;

- A Antonio Cammi, mio professore e relatore, mentore e psicologo, baluardo e rifugio al quale rivolgermi ogni qual volta ne avevo bisogno. A Antonio Cammi, la mia "Minas Tirith";

- A Laura Savoldi, mia professoressa e relatrice, per avermi sempre aiutato quando ne avevo bisogno, durante tutto il mio percorso di studi magistrali. A Antonio Cammi, la mia "Gandalf";

- A Stefano Lorenzi, ricercatore sempre disposto ad aiutarmi, a prescindere dai suoi impegni. Ogni volta che ho bussato alla porta del suo ufficio, la risposta è sempre stata "Ciao, dimmi";
- a Piero Ravetto e Mario De Salve, miei professori, pozzi di conoscenze e di vita, che con i loro consigli mi hanno guidato durante il mio percorso di studi;
- A tutto il NEMO group del Politecnico di Torino. Ad Antonio Froio, che più volte ha speso pomeriggi interi per risolvere problemi informatici, permettendomi di lanciare delle simulazioni sulle risorse informatiche; ad Andrea Zappatore, per il supporto datomi ad Aprile; ad Alberto Brighenti e Nicolò Falcone, per le divertenti discussioni avute in ufficio;
- A Nicolò Abrate, compagno di classe, amico fidato e da poco dottorando del NEMO group. Non dimenticherò mai tutto quello che hai fatto per me, dai corsi di fusione, ai files latex, ai "vari consigli". Grazie, Nico, grazie davvero;
- A Alessandro Pantano, compagno di classe, amico fidato, insieme al quale mi sono avvicinato per la prima volta al mondo della CFD, alle nottate insonni, alla depressione più totale, ai «ma che sono questi grafici?!» , al tubo triangolare dentro il quale scorre acqua in moto turbolento (dimmi, lettore, hai mai visto in vita tua un c***o di tubo triangolare? Se lo dico a «Geppo» , l'idraulico del paese in cui vivo, mi sputa in un occhio!);
- A tutto il Nuclear Group del Politecnico di Milano. A Carolina Introini e Eric Cervi, per la miriade di consigli e aiuti che mi hanno dato su OpenFOAM; a Tommaso Barani, Marco Cauzzi, Andrea;
- A Parikshit, mio mentore di Comsol Multiphysics, sempre gentile, sempre disponibile, sempre pronto ad aiutarmi;
- A tutti gli amici del quarto piano del collegio Einaudi e, in particolare, ad Alessandro Soli, mio insegnante di OpenFOAM nei primi mesi in cui mi approcciavo a questo ignoto software (che by the way mi metteva un chilo di aglio nella carne mentre cucinavo ed ero distratto), e a Mirko Filice, «il mio pupillo» ;
- A tutti gli amici della residenza Newton, ed in particolare al mitico Giorgione, a Matteo, a Aliosha (Alessio), che semplicemente è lo zar, a Paride e a Biagio. Mi avete sfamato in maniera disinteressata, senza avvelenarmi, duranti i giorni di reclusione a Luglio, quando scrivevo l'ultima parte di questo masterpiece/-porcheria (Aliò, tu niente hai fatto, solo la vongolata una volta! Però, lo ammetto, a cucinare sei il migliore);
- A tutte le coinquiline con cui ho convissuto quest'anno, a Giusy, a Giuliana e pure a Claudia. Ma soprattutto, a Noemi, l'unica costante nel porto di mare che è via ***;

- A te, che pur senza far nulla mi sei sempre stata vicina nel corso del mio percorso di tesi magistrale;
- A mio cugino, per tutto, per sempre;
- A mia madre ed a mio padre, colonne portanti che hanno avuto e continuano ad avere un ruolo fondamentale nella mia vita; a loro, che mi hanno sostenuto e difeso a spada tratta per tutto il percorso di studi universitario. Grazie dei consigli, degli aiuti, delle parole di conforto, delle "seccature". Grazie per esserci sempre stati e per esserci ancora;
- A mio fratello, che mentre scrivo queste righe mi appare non insieme a Nala, non mentre arrampica, non in montagna, suo vero amore, ma seduto sulla sedia nella cucina di nonna, alle tre del pomeriggio del 7 Ottobre 2015, quando fuori c'era il sole, ma io vedevo solo buio;
- A tutti quelli che non mi sono venuti in mente ora e che ho dimenticato di ringraziare, ma che meriterebbero ben più di due semplici righe;
- E poi, se mi si permette... A me stesso. Per aver sempre stretto i denti e non essermi mai arreso del tutto, per non aver mai mollato, anche quando tutto sembrava perso, anche quando non vedevo che nero nel tunnel, nonostante l'aiuto forte e costante delle persone a me care. A me che, quando (semmai) rileggerò la mia tesi, tra trent'anni, potrò ricordare di aver sofferto tanto, di aver sputato sangue, ma, nonostante tutto, di esserci riuscito e di avercela fatta.

«Questa conclusione, benché trovata da povera gente, c'è parsa giusta, che abbiam pensato di metterla qui, come il sugo di tutta la storia.

La quale, se non v'è dispiaciuta affatto, vogliatene bene a chi l'ha scritta (Giuseppe), e anche un pochino a chi l'ha raccomandata (muà - leggere come «moi» in francese). Ma se in vece fossimo riusciti ad annoiarvi. credete che non s'è fatto apposta.»