Universidade do Minho
Escola de Engenharia

João Luís Oliveira Pedro

Numerical Modelling of the Filling Stage of the Injection Moulding Process

Janeiro de 2020

Universidade do Minho
Escola de Engenharia

João Luís Oliveira Pedro

Numerical Modelling of the Filling Stage of the Injection Moulding Process

Dissertação de Mestrado
Mestrado Integrado em Engenharia de Polímeros

Trabalho efetuado sob a orientação do
Professor Doutor Célio Bruno Pinto Fernandes
Professor Doutor João Miguel de Amorim Novais da Costa Nobrega

Janeiro de 2020

Acknowledgments

This Master Thesis is the culmination of 5 years of experiences, that made me grow and learn a lot, so I could not fail to thank all those who accompanied me along this journey.

To Célio Fernandes and Professor João Miguel Nóbrega for their excellent guidance, help, motivation, and patience throughout this year of work. No doubt that without the help of both I would not be able to achieve these results.

To Luís Martins for the fact that he let me use his work equipment and helped me with the various doubts with Moldex3D®. If I managed to obtain the results present in this thesis, it was also due to your availability.

Also thank the entire office for the integration, coexistence, and for the people and friendships that I made this last year. Special thanks to Ricardo Costa for his patience and help with LaTeX, without a doubt that I would not be able to learn to work with the program so quickly without it.

I would like to acknowledge the funding by FEDER funds through the COMPETE 2020 Programme and National Funds through FCT - Portuguese Foundation for Science and Technology under the projects UIDB/05256/2020-UIDP/05256/2020 and MOLDPRO – Aproximações multi-escala para moldação por injeção de materiais plásticos (POCI-01-0145-FEDER-016665). I would like also to acknowledge the support of the computational clusters Search-ON2 (NORTE-07-0162-FEDER-000086) and Minho Advanced Computing Center (MACC).

Finally, two more special thanks to those who have always been there from the beginning. Firstly, to the people who accompany me from the first day of this experience, and who contributed to making these years unforgettable, namely, Mafalda, Catarina, Cíntia, Carina, Afonso, Piairo, Nuno, and Bruno. To them, thank you very much for all the good and less good moments, and for everything I learnt from you.

In conclusion, the most important thanks of all, to my family, but especially my parents and brother, because for every victory achieved, you need a great base, and they are my basis from the beginning.

Modelação numérica da Fase de Enchimento do Processo de Moldação por Injeção

Resumo

O aumento da procura dos materiais plásticos promoveu um crescimento na procura e exigência de vários processos de transformação, como acontece com a moldação por injecção, que é hoje em dia o processo industrial de transformação de materiais plásticos mais utilizado. Esta técnica de transformação apresenta os melhores retornos financeiros a médio/longo prazo e, uma elevada gama de geometrias que podem ser produzidas. É um processo que envolve um elevado custo fixo não só pelos moldes utilizados, mas também pelas muitas variaveis de processo que são necessárias controlar e otimizar para se atingirem as condições ótimas de processamento. É exatamente devido a este tipo de requisitos que a simulação passou a ganhar mais importância, sendo hoje em dia etapa indispensável no processo de conceção, tanto no controlo das variáveis do processo, como na otimização do mesmo, com o objetivo de se atingir níveis de eficácia elevados. Este trabalho trata aspectos relacionados com a modelação numérica do processo de moldação por injecção de termoplásticos, e teve como principais objectivos a validação de um código open-source desenvolvido para a simulação do processo, e a criação de um caso de estudo de sucesso que pudesse ser replicado experimentalmente. A principal motivação para a realização do mesmo prende-se com o facto de que até aos dias de hoje, não existem no mercado dos softwares de distribuição livre, códigos capazes de simular o processo com as desejadas características de precisão e com tempos de cálculo adequados ao ritmo de produção industrial. Primeiramente este trabalho teve por objetivo identificar as equações que governam as diferentes fases do processo de moldação por injecção e os métodos de cálculo empregues em códigos de modelação numérica. Nesse contexto, foram também estudados os modelos constitutivos adequados para materiais plásticos. Em seguida, com o recurso a diversos casos de estudo, foram efetuados testes de validação ao solver open-source denomindado, *openInjMoldSim*, desenvolvido com base na biblioteca computacional OpenFOAM®. Numa primeira etapa verificou-se que os perfis de velocidade e pressão calculados com o solver se assemelhavam aos obtidos com equações analíticas. No segundo caso, foram empregues duas formulações da equação de estado, uma compressível e outra incompressível, cuja comparação permitiu concluir que as formulações incompressíveis melhoram o tempo de cálculo e a estabilidade do processo numérico, sem afetarem significativamente a precisão. Por fim, fez-se um estudo de comparação entre o solver open-source e o software comercial Moldex3D®. Deste estudo, conclui-se que para a mesma precisão, e não tendo total conhecimento das equações e métodos de cálculo empregues no software comercial, o Moldex3D® foi 10 vezes mais rápido do que o OpenFOAM®, quando se utilizam o mesmo número de processadores. No entanto, os tempos de cálculo passam a ser idênticos se todo o potencial de paralelização do OpenFOAM® for aproveitado. **Palavras-chave:** Equação de estado, Fase de injeção, Modelação numérica, Moldação por injeção, Moldex3D®, OpenFOAM®.

Numerical Modelling of the Filling Stage of the Injection Moulding Process

Abstract

The increased demand for plastic materials promoted an increase in the requirements of various transformation processes, namely injection moulding, which is nowadays the most used transformation process for the manufacture of plastic material parts. This transformation technique presents the return of investment in the medium / long term and a wide range of geometries that can be produced. It is a process that has a high fixed cost not only due to the moulds used, but also due to the many process variables that are necessary to control and optimize to achieve the optimal processing conditions. It is exactly due to this type of requirements that the simulation started to gain more importance, being nowadays an indispensable step in the design phase, either in the control of the process variables or in the optimization of the same, to reach high levels of effectiveness.

This work deals with aspects related to the numerical modelling of the thermoplastic injection moulding process, and its main objectives were the validation of an open-source solver developed for the simulation of the injection moulding process, and, the creation of a successful case study that could be replicated experimentally. The main motivation for this is due to the fact that there are not any codes in the free distribution software market capable of simulating the injection moulding process with the desired precision characteristics, and with calculation times appropriate to the industrial production pace.

Firstly, this work aimed to identify the equations that govern the different phases of the injection moulding process, and the numerical methods used in numerical modelling codes. In this context, the constitutive models suitable for plastic materials were also studied. Then, using several case studies, validation tests were carried out using the open-source solver named, *openInjMoldSim*, which is based on the OpenFOAM® computational library. In a first step, it was found that the velocity and pressure profiles predicted by the solver were similar to those obtained with analytical solutions. In the second case, two formulations of equation of state were used, one compressible and the other incompressible. The results obtained allowed to conclude that the incompressible formulation reduce the calculation time and improve the stability of the numerical process, without any significant loss of accuracy. Finally, a comparison study was carried out between the open-source solver and the commercial software Moldex3D®. From this study, we concluded that for the same precision, and not having full knowledge of the equations and calculation methods used in commercial software, Moldex3D® was 10 times faster than OpenFOAM®, when using the same number of cores in the computation. Moreover, when taking full advantage of the parallelization capabilities of OpenFOAM®, the calculation time was the same in both softwares.

Keywords: Equation of state, Filling stage, Injection Moulding Process, Moldex3D®, Numerical Modelling, OpenFOAM®.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Table of Contents

List of Figures

List of Tables

List of Symbols

Latin symbols

A	Area [m^2]
$\underline{a}$	Vectorial quantity
a_T	Time-temperature shift factor
c_p	Specific heat capacity [$\mathrm{J}/(\mathrm{kg}\cdot\mathrm{K})$]
$\underline{\underline{D}}$	Rate of strain/deformation tensor [$1/\mathrm{s}$]
$\underline{d}$	Vector connecting cells P and N
E	Activation energy [kJ/mol]
e	error [%]
F	Mass flux [kg/s]
f	Face
f_x	Interpolation factor
h	Convection heat transfer coefficient [$\mathrm{W}/(\mathrm{m}^2\cdot\mathrm{K})$]
I	Invariants
II_A	Second invariant of matrix A
$\underline{\underline{I}}$	Identity matrix
$\underline{k}$	Non-orthogonal contribution vector [m^2]
k	Thermal conductivity [$\mathrm{W}/(\mathrm{m}\cdot\mathrm{K})$]
L	Length [m]
$\underline{\underline{L}}$	Velocity gradient tensor [$1/\mathrm{s}$]
m	Consistency of power-law model [$\mathrm{Pa}\cdot\mathrm{s}^{\mathrm{n}}$]
N	Neighbour cell
$\underline{n}$	Normal vector to a surface
n	Power-law index
o	Apparent order
P	Cell of interest
p	Pressure [Pa]
p_m	Pressure from fluid motion [Pa]
p_t	Thermodynamic pressure [Pa]
$\underline{q}$	Heat flux [W/K]
R	Grid refinement factor
r	radius [m]
S_ϕ	Source term
S_p	Source term linear coefficient
S_u	Constant part of source term
$\underline{S}$	Face area vector [m^2]
T	Temperature [°C]
T_g	Glass transition temperature [°C]
T_m	Mould temperature [°C]
T_{melt}	Melt temperature [°C]
$\underline{t}$	Stress vector [Pa]
t	Time [s]

t_{SO}	Switch-over time [s]
U	Velocity magnitude [m/s]
$\underline{u}$	Velocity vector [m/s]
$\underline{u}_r$	Artificial velocity term [m/s]
V	Volume [m^3]
$\widehat{V}$	Specific volume [m^3/kg]

Greek Symbols

α	Polymer volume fraction
β	Coefficient of volumetric expansion [1/°C]
Γ_ϕ	Diffusion coefficient
$\dot{\gamma}$	Shear rate [1/s]
Δt	Time interval [s]
$\underline{\Delta}$	Orthogonal contribution vector [m^2]
η	Dynamic viscosity for non-Newtonian Fluids ($\eta = f(\dot{\gamma})$) [$\mathrm{Pa} \cdot \mathrm{s}$]
η_0	Viscosity at a null shear rate [$\mathrm{Pa} \cdot \mathrm{s}$]
η_∞	Viscosity at an infinite shear rate [$\mathrm{Pa} \cdot \mathrm{s}$]
ι	Isothermal compressibility coefficient [1/Pa]
λ	Relaxation time [s]
λ_i	Eigenvalues
μ	Dynamic viscosity for Newtonian fluids [$\mathrm{Pa} \cdot \mathrm{s}$]
ρ	Density [$\mathrm{kg/m}^3$]
$\underline{\underline{\sigma}}$	Stress tensor [Pa]
$\underline{\underline{\tau}}$	Deviatoric stress tensor [Pa]
τ^*	Shear stress at the transition between Newtonian and shearthinning regions [Pa]
ϕ	General scalar quantity
χ	Pressure dependent coefficient [Pa^{-1}]
∇	Gradient vector [1/m]

Superscripts/subscripts

$\overline{\Box}$	Average value
$\Box^0$	Value of the previous time-step
$\Box^T$	Transpose operation
$\Box^n$	Value of the current time-step
$\Box_0$	Initial value
$\Box_N$	Value at cell center of N
$\Box_P$	Value at cell center of P
$\Box_{ext}$	Extrapolated value
$\Box_f$	Face value
$\Box_i$	value for a specific level of refinement

Constants

$\mathrm{R_g}$	Ideal gas constant ($\mathrm{R_g} = 8,314$ [$\mathrm{J/mol \cdot K}$])
$\underline{\mathrm{g}}$	Gravitational acceleration vector ($\lVert \underline{g} \rVert = 9.81$ [$\mathrm{m/s}^2$])

1. Introduction

This chapter provides an overview of this work and its organization. Firstly, the motivations that led to the selection of the topics covered in this thesis are presented. Then, bearing in mind the work motivations, the main objectives are introduced, and finally, the organization/structure of the thesis is given.

1.1. Motivation

Injection moulding process is the main industrial process for the manufacturing of plastic products because it allows a high production rate, and obtaining a wide range of geometries, with high-quality characteristics. However, it is also a high-cost process, especially because of the required moulds, which must present large mechanical robustness and are usually difficult to machine, due to their complex shape. During the injection moulding process there is the need of controlling a huge number of variables and process conditions to achieve the best possible results, meaning that the injected moulded part has good performance and aesthetic characteristics, which is several times achieved by experimental trial-error approaches, and, consequently, it involves large costs. This way, numerical simulation can play an important role to guide process improvements, being nowadays a powerful tool, often used in industry to support design activities.

In the framework of the injection moulding process, the most disseminated numerical simulation software presents two major drawbacks: they are proprietary and its difficult, or even impossible, to adapt the code to specific needs. The first aspect means that companies need to pay licenses to use the softwares, which usually presents a high cost, thus, limiting its use in small to medium size companies. The second aspect indicates that the user is not able to change the code to try a different approach or adapt it to its needs, because there is not access to the source code.

These limitations led the injection moulding community to look for open-source code alternatives, which gave rise to different numerical implementations, as the ones provided by *Billy Araújo* [1], *Kristjan Krebelj* [2], or *Joszef Nagy* [3]. Two of those cotributions were implemented in OpenFOAM® [4], which give the capability of exploring different computational problems in varied fields, ranging from simple flows in a channel till simulations of aerodynamics in the automotive area, using several constitutive models. Injection moulding still is a process under investigation in that computatinal library, with many individuals trying to create valuable alternatives to proprietary software into the OpenFOAM® [2, 3]. Today, although the number of users of open-source codes is growing, there is still some barriers that need to be broken, to disseminate these codes starting by assessing code predictions, and, if required, improving the performance and precision of the developed solvers. This could be achieved by the presentation of some successful applications or case studies.

The main motivation behind this work is the desire of in the future be able to create an open-source solver for injection moulding in OpenFOAM® [4], at least, with similar capabilities to the proprietary codes that could be used not only by bigger companies but the smaller ones, which could be a major contribution to the field and its improvements.

1.2. Objectives

The main objectives of this work were to assess in detail *openInjMoldSim*, a solver developed in OpenFOAM® [4] by Kristjan Krebelj [5–7] for the numerical modelling of the injection process, and to create a successful case study that could be replicated experimentally in the future. As a long term goal of developing a solver for the injection moulding process, first, it is necessary to identify the equations that govern the different phases of the process, and, the calculation procedure employed in numerical modelling codes. Furthermore, the constitutive models suitable for plastic materials were also investigated. After identifying these topics, a comparison between the literature and the solver was made to verify the calculation procedure employed in the *openInjMoldSim*.

Subsequently, to validate and assess the open-source solver three cases studies were employed. The first case study aimed to verify whether the velocity and pressure profiles obtained with the open-source solver were similar to those obtained through analytical solutions. In order to gain more knowledge of the process, two variations of the equation of state were employed (compressible and incompressible), and the results obtained for the different levels of mesh refinement were compared. Finally, a comparison was made in terms of precision and performance of the results obtained in a 3D case study typical of the injection moulding process, using both, *openInjMoldSim*, and Moldex3D® [8], a commercial program widely used in the injection moulding industry. The comparison of both numerical approaches were made by applying the same initial and boundary conditions in both softwares, which might be a difficult task in the commercial one because of the above-mentioned difficulty in adapting the code to our specific needs.

1.3. Thesis organization

The methodology and results obtained in this work are presented in seven different chapters, with the following contents. The **first chapter**, this one, is the introduction where, the motivations and objectives of the work are presented. The **second chapter** describes the theoretical background, where the basic information about polymers and injection moulding process are explained, as well as, the models that suit material's main characteristics for the filling stage of the injection moulding process. This way, the **second**

chapter serve as basis of all the subsequent information given in this work. The **third chapter** comprises the State of Art, an overview of what has already been achieved, and what is yet to be developed in the area of numerical modelling of the injection moulding process. Subsequently, in the **fourth chapter** the mathematical foundations to model the injection moulding process, as well as a description on how these models can be implemented numerically are presented. In the **fifth chapter**, the cases of study chosen to make the numerical studies are presented, namely, geometries, meshes, materials, and initial and boundary conditions used in the calculations. The **sixth chapter** present the results analysis of the selected cases studies. The analysis comprises, in a first phase, velocity and pressure profiles assessment, then a study on the equation of state of the polymeric materials, and, finally, the comparison of the results obtained with a commercial software and an open-source software in terms of performance and accuracy. In the **seventh chapter**, the main conclusions and proposals for future works are addressed.

2. Theoretical Background

This chapter presents the theoretical foundations about the main fields covered in this Master thesis, especially in what concerns to the polymers rheological and physical behaviours and the injection moulding process. First, a review about polymers and their characteristics will be presented, and, subsequently a general description of the injection moulding process will be given. The remain sub-sections provide an overview of the main constitutive models that are relevant for polymeric materials. The main objective of this chapter is to give a complete review of the base theoretical background required to understand the numerical modelling approaches, presented in the subsequent chapters, and its applications to the injection moulding process.

2.1. Polymers

Polymers are high molecular weight compounds, with long chains built of many similar repeating units. The repeating units are connected through covalent bonds over all molecular structure, and with additional chemical groups in their structure, which, associated to their arrangements, mark the difference between the range of polymers [9]. These repeating units can be near a million and are usually organized in a disorderly entangled cloud, which promote a high viscosity (usually between $10^2 - 10^6$) (Pa.s), *i.e* polymers present a high resistance to the flow [10]. In injection moulding, there are two main groups of plastics used for processing, the thermoplastics and the thermosets, which main difference is the ability of the last to cross-link [11].

The thermosets are polymers that show cross-linking reaction between chains through branch groups, resulting in a rigid three-dimensional lattice structure [12]. The reaction takes place with temperature increase, radiation, or even chemical agents, upon which, they cannot be remelted again [13]. The thermoplastics are polymers that soften with heating and solidify with cooling, and their structure does not cross-link [9]. The main feature of these materials is that repeated cycles of heating and cooling can be performed many times without significant changes in properties [14]. This study focus on thermoplastics polymers because, from an industrial perspective, they are more used in injection moulding applications than thermosets.

The thermoplastic group can be divided into amorphous and semi-crystalline polymers based on their ability to develop crystalline structures [15]. The amorphous polymers molecular chains, at melt state, acquire mobility and adopt an entangled disordered organization like a cloud, known as rubbery state [11]. With the cooling, the material changes from the rubbery state to a hard glass-like appearance, maintaining the disorganization of its molecular chains [9]. The temperature at which this phenomenon occurs is called glass transition temperature, T_g [12]. At the melt state, the semi-crystalline polymers present the same characteristics of the amorphous polymers [16]. When cooling takes place, their molecules arrange themselves into well organized 3D structures which led to the formation of crystalline regions [12]. As this process of crystallization is not uniform all over the molecules chains, zones where organized structures do not exist are formed, and, consequently, amorphous regions are also present, which is the motivation for the semi-crystalline designation. Due to the existence of crystalline regions, these polymers only become melted when the crystalline regions are destroyed, at temperatures above the melt temperature, T_m, thus, they present both T_g and T_m [17]. Due to the presence of crystalline regions, semi-crystalline are opaque to light transmission, contrarily to the amorphous that are translucent [16].

The different properties presented by polymers play an important role in the final characteristics of injected products. Additionally, these properties change during the process, and therefore, its determination is not a simple task [12]. Thus, in the remaining of this chapter, the properties of polymers that are needed to model the filling stage of the injection moulding process will be discussed along with its main models. Therefore, for the numerical modelling of the filling stage of the injection moulding process of thermoplastics, [12] there are properties of interest that need to be well predicted, namely:

- viscosity, which varies with shear rate, temperature and pressure;
- specific heat capacity, which depends on temperature;
- thermal conductivity, which depends on temperature;
- material density, which vary with temperature and pressure;

2.2. Injection Moulding Process

From all thermoplastic processing techniques, injection moulding process is the most important one [15]. As illustrated in Figure 1, in general terms, the process consists of forcing a melted material to fill a mould cavity, which presents a geometry similar to the final part. In more detail, initially the hopper is filled with the polymer granules, which, by gravity, flow towards the barrel, and due to the rotation of a screw, the granules are heated and melted to a molten state (like a paste), simultaneously, the screw is forced to move back (1. Plasticization stage, Figure 2). Afterwards, due to the advance of the screw, the polymer melt is forced to enter the mould (2. Filling stage, Figure 2) [18]. Inside the mould, the material fills the cavity that has final part geometry, and then after cooling and, consequent, solidification (3. Cooling stage, Figure 2), the final part is ejected (4. Ejection stage, Figure 2) [15].

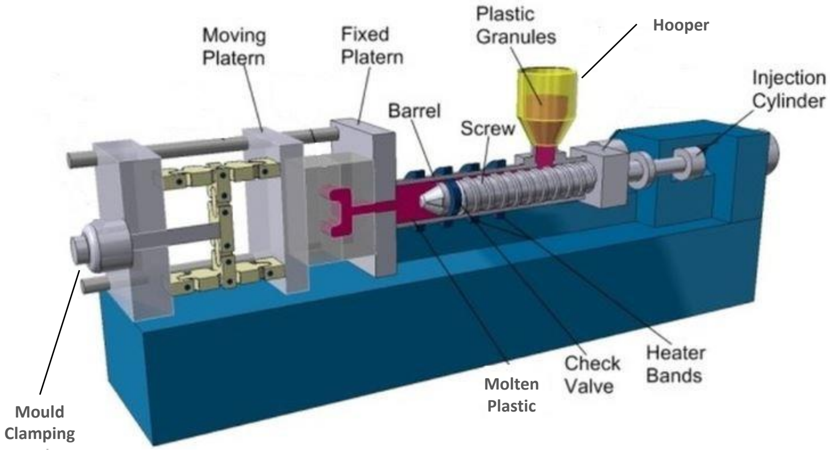

Figure 1: Injection moulding machine components (Image adapted from [19]).

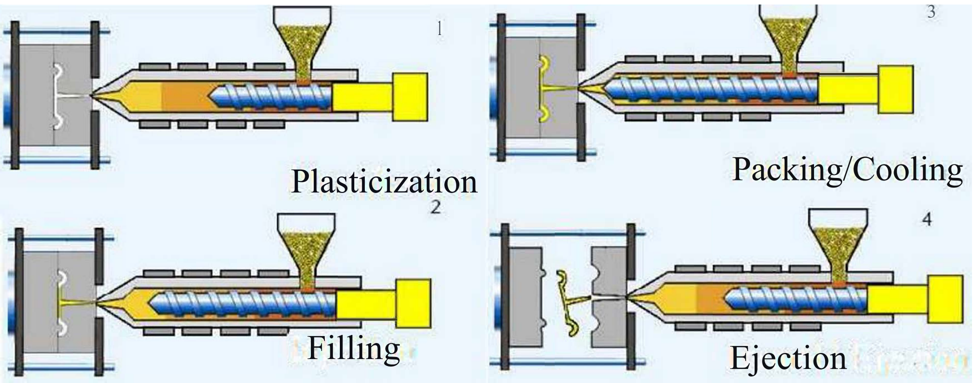

Figure 2: Injection moulding process phases (Image adapted from [20]).

In the filling phase, another phenomenon that might happen is viscous dissipation, which is characterized by an increase in the temperature of the melt due to the shear rates applied to the material [21]. After the filling, there is the packing phase, where the flux is almost null and the material starts to cool down. Therefore, shrinkage might happen, thus, to prevent it, the injection machine is still injecting some material into the mould at high pressure [11]. After the packing phase, starts the longer phase of the cycle (see Figure 3) where cooling/solidification happens. The principal mechanism of heat transfer is by heat conduction to the mould but polymers present very low thermal conductivity, usually they are known as insulators, consequently, the cooling phase takes a significant part of the cycle, as Figure 3 shows [21].

Due to the easy automation, injection moulding is a process that allows large scales of production, especially for small and thin parts. Although it allows producing geometrical complex parts, it is usually restricted to low thickness parts due to the, above mentionated, low thermal conductivity characteristic [15]. As stated before, the polymers present a high viscosity, and, when melted, they exhibit a high resistance to flow, which results in very high pressures during the injection moulding process [15]. Due to this characteristic, the injection moulds need to be robust, which means that the moulds are very expensive, and consequently, it is a high cost process. This way companies frequently appeal to modelling to simulate and optimize the process instead of the experimental trial-error method [12].

Being the objective of this study to validate *openInjMoldSim*, a solver developed in OpenFOAM® by Kristjan Krebelj [2] for the numerical simulation of injection moulding process, it is important to understand the details of the physical phenomena that occur during the injection moulding process, namely the mathematical governing equations that are the ground basis of it, which will be presented in Chapter 4.

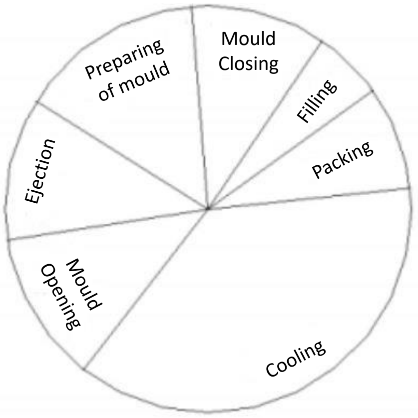

Figure 3: Injection moulding cycle with the average times of each phase (Image adapted from [22]).

2.3. Material Constitutive Relationships

The relation between the stress exerted by the material and the deformation rates imposed during the flow is essential to numerical modelling [23]. The material rheological properties dictate the way it deforms and flows in response to the applied forcings, as well, the evolution of the induced stresses [11]. As stated before, the viscosity, thermal properties and material density determine the ability the melt has to fill the mould, *i.e*, the pressure required to fill the cavity [12].

Polymers are viscoelastic materials, which means that they combine properties of viscous fluids and elastic solids. In the first case, the energy promoting deformation is dissipated, and, in the second case, the deformation energy is stored [11]. These materials require time for the strain to accommodate the applied stress field. The time constant responsible for this behaviour is known as relaxation time, λ, which is a material characteristic [24]. Furthermore, this means that the deformation history plays a role on the stress state for these materials [25].

The relations between time dependencies, strain and stress are ruled by a constitutive equation [24]. The material constitutive law can be expressed either explicitly or implicitly, depending on the ratio between the deformation rate and the material time scales, *i.e*, whether the deformation is slow enough to allow neglecting the material time dependence. Furthermore, as polymers are characterized for having long molecular chains, the time that takes the strain to accommodate stress may reach several seconds, meaning that the slower the deformation rate, the less relevant time dependecy is [24].

The characterization of the stress in a point of a body requires a second order tensor, which comprises nine components, as illustrated in Figure 4 [26].

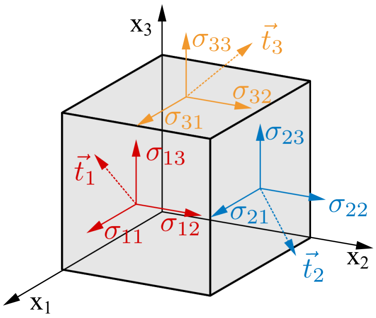

Figure 4: Stress components (Image adapted from [27]).

The stress tensor is usually equated as:

$$\underline{\underline{\sigma}} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}. \tag{1}$$

To assure a null angular momentum balance the stress tensor must be symmetric, *i.e*, $\sigma_{ij} = \sigma_{ji}$ [12]. At a point of interest, the relation between $\underline{\underline{\sigma}}$ and the stress vector $\underline{t}$, which characterizes the stress in a specific plane that intersects the point, and is normal to $\underline{n}$, is given by

$$\underline{t} = \underline{\underline{\sigma}} \cdot \underline{n}, \tag{2}$$

where $\underline{n}$ is the plane normal vector. For a fluid at rest, the stress tensor is equal to the thermodynamic pressure, p_t [12]. This pressure is taken from the equation of state (Section 2.5), which relates the pressure to the material's specific volume and temperature. In those conditions, the stress tensor is given by

$$\underline{\underline{\sigma}} = -p_t \underline{\underline{I}}. \tag{3}$$

where $\underline{\underline{I}}$ is the identity matrix given by

$$\underline{\underline{I}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

When there is flow additional stress contributions must be considered and the stress tensor becomes

$$\underline{\underline{\sigma}} = -p_t\underline{\underline{I}} + \underline{\underline{\tau}}, \tag{5}$$

where $\underline{\underline{\tau}}$ is the extra or deviatoric stress tensor. In Equation (5), p_t is the isotropic part of $\underline{\underline{\sigma}}$, acting in all normal directions, $p_t = (\sigma_{11} + \sigma_{22} + \sigma_{33})$, thus, $\tau_{11} + \tau_{22} + \tau_{33} = 0$ [28].

Material motion comprises translation represented by the velocity vector, $\underline{u} = (u, v, w)$, where u, v, and w are the velocity components in each direction x, y, and z respectively, deformation and rotation [12]. The last two are dependent on the velocity gradient tensor, which is given by

$$\underline{\underline{L}} = (\nabla\underline{u})^{\mathrm{T}} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial u}{\partial z} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial u}{\partial z} \end{bmatrix} \tag{6}$$

The superscript T in Equation (6) indicates the transpose operation. The rate of deformation (or strain) tensor, $\underline{\underline{D}}$, is a function of the $\underline{\underline{L}}$, defined as [12]

$$\underline{\underline{D}} = \frac{1}{2}\left(\nabla\underline{u} + (\nabla\underline{u})^{\mathrm{T}}\right) = \frac{1}{2}\left(\underline{\underline{L}}^{\mathrm{T}} + \underline{\underline{L}}\right). \tag{7}$$

The rheological behaviour of different materials is characterized by its constitutive laws, which for fluids relates $\underline{\underline{\sigma}}$ and $\underline{\underline{D}}$. There are different laws appropriate for distinct fluids. The most relevant ones employed for polymeric materials will be described in the following subsections.

2.3.1. Types of fluids

For Newtonian fluids the stress is directly proportional to shear rate, being the constant of proportionality called viscosity μ, which is constant at a given temperature, and therefore, does not depend on $\underline{\underline{D}}$ [28]. These fluids have a particularly simple constitutive equation that relates the extra stress tensor with the rate of deformation tensor in the form of

$$\underline{\underline{\tau}} = 2\mu\underline{\underline{D}} - \underbrace{\frac{2}{3}\mu\,(\nabla\cdot\underline{u})\,\underline{\underline{I}}}_{p_m\underline{\underline{I}}}. \tag{8}$$

The last term of the equation is isotropic, like a pressure, and results from the fluid motion and its compressibility, therefore it will be denoted as $\mathrm{p_m}\underline{\underline{I}}$ [12]. For an incompressible fluid, which means that density, or specific volume, does not change with pressure and temperature, the divergence of the velocity vector is zero, $(\nabla\cdot\underline{u} = 0)$. Then, Eq. (8) becomes

$$\underline{\underline{\tau}} = 2\mu\underline{\underline{D}}, \tag{9}$$

And recalling Eq. (5) we obtain the constitutive equation for a Newtonian fluid,

$$\underline{\underline{\sigma}} = -\left(p_t + p_m\right)\underline{\underline{I}} + 2\mu\underline{\underline{D}} = -p\underline{\underline{I}} + 2\mu\underline{\underline{D}}. \tag{10}$$

However, several fluids, has happens with polymeric materials, present a different behaviour, and are designated non-Newtonian fluids [12]. Based on the relation between rate of deformation and time dependence, non-Newtonian fluids can be divided into inelastic (no time dependence) or elastic (time dependent) fluids [24]. For inelastic fluids, the viscosity can be a function of temperature, T, pressure, p, and shear rate, $\dot{\gamma}$, as stated in Equation (11), being in this case designated by generalized Newtonian fluids (GNF) [12].

$$\underline{\underline{\tau}} = 2\eta\left(T, p, \dot{\gamma}\right)\underline{\underline{D}}. \tag{11}$$

The time dependence originates a second group called elastic fluids or viscoelastic fluids, where polymers belong, as stated before [25]. In fact, the time dependence of the viscoelastic materials only matters when the time-frame of the process is long enough to allow neglecting the time-frame of the material [24]. This means that, for injection moulding, and especially during the filling stage, which is the quickest phase of the process, the phenomena dependent of the material time frame have some importance. Although, as the computational resources increase with a viscoelastic model, in proprietary software, the GNF model is employed because it is computationally less expensive and in many cases it allows accurate enough results [8, 28].

For GNF, the viscosity is function of $\eta\left(T, p, \dot{\gamma}\right)$, thus the relations between η and T, p and $\dot{\gamma}$ must be identified. It is not possible to establish a direct relationship between the η and $\underline{\underline{D}}$, because the components of the latter change depending on the coordinate system considered, and the η is a scalar, meaning that it remains constant whatever the coordinate system considered. However, every second-order tensor has three independent scalars designated as invariants because their values are independent of the coordinate system considered. Therefore, the relation is established with the invariants of rate of deformation tensor $\underline{\underline{D}}$. The first invariant of $\underline{\underline{D}}$ is zero for incompressible fluids [29], and the third one is also zero for shear dominant flows [29], then, the relation will be established between η and the second invariant of $\underline{\underline{D}}$, which accounts for tensor magnitude [28]. The definition of the first, second and third invariants is presented in the Appendix A. The second invariant of a tensor $\underline{\underline{A}}$ is given by the double dot operation, that is detailed in Appendix A, and has the form of

$$II_A = \underline{\underline{A}} : \underline{\underline{A}}. \tag{12}$$

For all flows, the magnitude of the strain rate tensor is given by

$$\dot{\gamma} = \sqrt{2II_{\underline{\underline{D}}}} = \sqrt{2\underline{\underline{D}} : \underline{\underline{D}}} = \sqrt{2tr\left(\underline{\underline{D}}^2\right)}, \tag{13}$$

where $\dot{\gamma}$ is usually designated by shear rate. Recalling Equation (8), for a GNF, the constitutive equation is given by

$$\underline{\underline{\tau}} = 2\eta\left(T, p, \dot{\gamma}\right)\underline{\underline{D}} - \underbrace{\frac{2}{3}\eta\left(T, p, \dot{\gamma}\right)\left(\nabla \cdot \underline{u}\right)\underline{\underline{I}}}_{p_m}, \tag{14}$$

Once again, p_m is neglected for incompressible flows. And, recalling Eq. (5) we obtain the constitutive equation for a GNF fluid

$$\underline{\underline{\sigma}} = -p\underline{\underline{I}} + 2\eta\left(\dot{\gamma}, T, p\right)\underline{\underline{D}}. \tag{15}$$

where $p = p_t + p_m$. On the subsequent sections the relations and models that relate η with $\dot{\gamma}$, T, and p will be discussed.

2.3.2. Viscosity models

In this sub-section, first, the shear-thinning behaviour of polymeric materials will be presented, which considers the dependency of η on $\dot{\gamma}$. After that, the effect of the temperature and pressure on viscosity will also be presented as well as the main models that govern all these relationships.

Effect of Shear Rate

The behaviour of a shear-thinning material, shear rate versus viscosity, presented by most of the polymers, is shown in Figure 5. With the increase of the shear rate, the viscosity decreases, meaning that the molecular orientation of the material and the entanglements are destroyed, and consequently the polymer flows easily [28]. Figure 5 depicts two plateaus and a transition region. The first plateau, called first Newtonian plateau, represented by η_0, is where the polymer chains are randomly coil and highly entangled, and the second plateau, called second Newtonian plateau, represented by η_∞ is where the molecules are highly aligned [25]. Several models can predict this behaviour appropriately, and they will be described hereafter.

Power law model

The power-law or Ostwald-de Waele [31, 32] model was developed in the twenties and represents the viscosity function proportionally to a power of the shear rate and has the form of

$$\eta = m\dot{\gamma}^{n-1}, \tag{16}$$

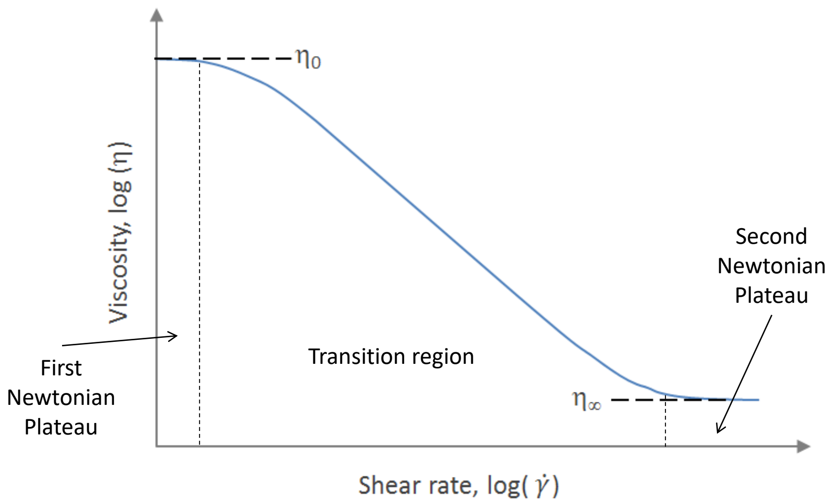

Figure 5: Shear-thinning behaviour of polymeric materials (Image adapted from [30]).

where m and n are material constants. The first is the consistency index and the second the power-law index. When $n = 1$ the Newtonian fluid relationship is obtained. Taking logarithms of both sides of Eq. (16), yields

$$\log\eta = (n-1)\log\dot{\gamma} + \log m, \tag{17}$$

thus, in the logarithm scale Equation (17), the power law represent a line with slope $(n-1)$, meaning that at high shear rates, similar to the ones found near the mould surface in the injection moulding process, the power-law model gives a very good representation of the polymer melt behaviour [29]. Although high shear-rates are presented near the mould surface, very low shear-rates exist at the centre of the channels and in all locations during the packing phase, and, therefore, this model does not give an accurate behaviour of the material in both situations [12]. This is the reason why power-law model is not used to simulate the injection moulding process [12]. Figure 6 shows the differences between the viscosity data measured for an HDPE (High-density polyethylene), and the viscosity prediction of power-law model [33]. Figure 6 shows that especially for low shear rates, the power-law model does not give a very good representation of the material's viscosity.

Carreau-Yasuda and Cross models

Both Carreau-Yasuda [34, 35] and Cross models [36] are able to predict the viscosity at lower shear rates better than the power-law because those models consider that value in an explicit manner. At the same time, both maintain the good agreement for high shear rates, and, therefore, both of them are the most used models in modelling viscosity of polymers as GNF [29]. Carreau-Yasuda and Cross models are represented by the following equations, respectively,

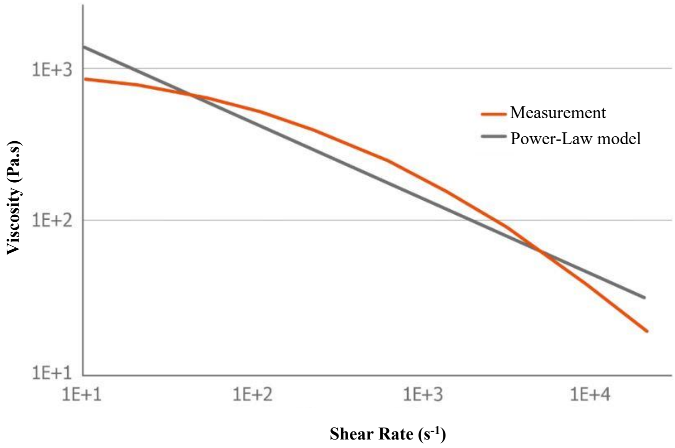

Figure 6: Representation of the power law rheological model compared with the shear-thinning behaviour of an HDPE. (Image adapted from [33])

$$\frac{\eta-\eta_\infty}{\eta_0-\eta_\infty}=[1+(\lambda\dot{\gamma})^a]^{\frac{n-1}{a}}, \tag{18}$$

$$\eta=\frac{\eta_0}{1+\left(\frac{\eta_0\dot{\gamma}}{\tau^*}\right)^{n-1}}. \tag{19}$$

where η_0 is the viscosity at null shear rate, η_∞ is the viscosity at the infinite shear rate, and n is the power-law index. For Carreau-Yasuda, λ is a material time constant parameter [11], and, when $a = 2$, value typical of many shear-thinning behaviours, the model reduces to Bird-Carreau model [28]. For Cross model, τ^* represents the shear stress at the transition from the Newtonian region (First plateau) to the Shear-thinning region.

These models have the advantage of combining both behaviours, this means that at very low shear rates, $\dot{\gamma} \rightarrow 0$, both predict the zero shear rate viscosity, η_0, and at high shear rates, it assumes "power-law" like behaviour [12]. Therefore, both of them are the most used models in modelling viscosity data for polymers, since their discrepancies are small, as shown in Figure 7.

When comparing both models, Hieber and Chiang [37], as well as W. Richards and D. Yoshimura [38] concluded that Cross model gives better results for the shear rate dependency. In their study, Hieber and Chiang [37] recognised that for five different polymeric materials, the deviations present in the values of viscosity predict by cross model were smaller than the ones found in Carreau-Yasuda model. W. Richards and D. Yoshimura [38] reached the same conclusions for polycarbonate resins. For this reason, the Cross model has been the most used model in proprietary software developed for injection moulding.

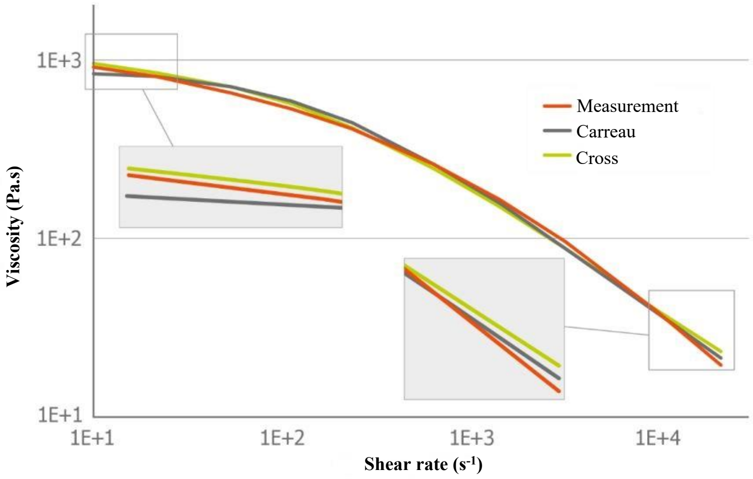

Figure 7: Representation of the Carreau-Yasuda and Cross rheological models compared with the shear-thinning behaviour an HDPE. (Image adapted from [33])

Temperature Effect

Temperature has a key role in viscosity because with its increasing, molecular mobility increases and, therefore, viscosity decreases, meaning that flow becomes easier [24]. This means that with the increase of temperature, the viscosity of the material decreases as Figure 8 shows, which presents the evolution of the viscosity with the shear rate for different temperatures for two different polymers (one amorphous (▲), and one semi-crystalline (●)) [12].

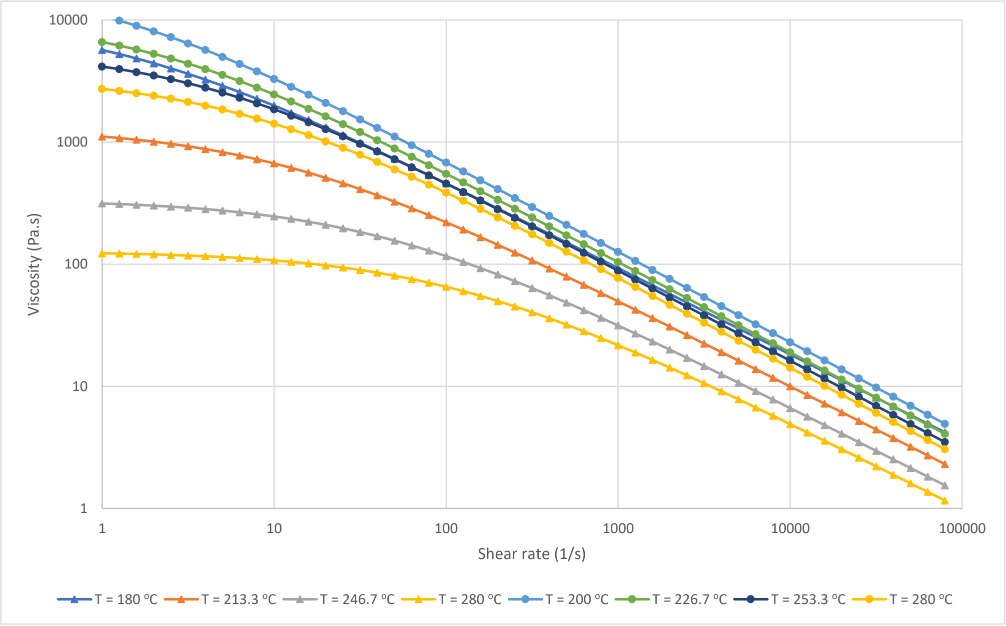

Figure 8: Viscosiy vs shear-rate for a polystyrene (▲) (amorphous) and a polypropylene (●) (semi-crystalline) at different temperatures.

As mentioned before, the viscosity for a GNF model needs to take into account the effect temperature has in viscosity modelling. For injection moulding process, temperature varies from the melt front (temperature range from $140°C$ to $300°C$ [21] for the most common thermoplastics) to the mould walls (usually between $45°C - 100°C$ [21], for the common thermoplastics), meaning high-temperature gradients are present [12]. To account for the effect of temperature on viscosity, the time-temperature superposition principle is considered [24]. This principle states that the viscosity at any temperature for a polymer can be obtained from the viscosity at temperature T_0 multiplyed by a shift factor $a_T(T)$ [29]. The thermo-rheological simplicity behind this relation is based on the fact that all molecular mechanisms involved have the same temperature dependencies [29].

The temperature dependence of the time-temperature shift factor, $a_T(T)$ may be represenred by different models depending on the temperature range, and the polymers we are working with. The main models for the effect of temperature in viscosity are the Arrhenius [39] equation and Williams-Landel-Ferry equation (WLF) [40].

Arrhenius equation

The relation for the shift factor of the time-temperature superposition model given by Arrhenius equation [39] has the following form

$$a_T = exp\left[\frac{E}{R_g}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right], \tag{20}$$

where E is the activation energy, R_g is the ideal gas constant, T_0 is the reference temperature in Kelvin, and T is the temperature of the material also in Kelvin [29]. The values of activation energy for some polymers are present in Table 1.

Table 1: Values of Activation Energy of Polymer Melts [41–43].

Polymers	Glass transition temperature (T_g $°C$)	E in $kJ \cdot mol^{-1}$ at 100°C above glass transition temperature
HDPE	-110	24-28
LDPE	-110	61-67
LLDPE	-110	29-32
Polypropylene	-10	36-50
Polystyrene	90-120	95-150
Polyamide 6	60	55-105
PMMA	90-110	167

Table 1 shows that the activation energy for polymers is the order of 10^{-4}J/mol. Furtermore, we conclude that with the increase of the stiffness of the molecular chains, the activation energy increases [29].

WLF equation

The WLF equation [40] is widely applicable in the rapid variation of viscosity in the range of temperatures between $T_g < T < T_g + 100$ [24, 29]. The shift factor a_T is given by

$$\ln a_T = -\frac{C_1^g\,(T - T_g)}{C_2^g\,(T - T_g)} \tag{21}$$

where T_g is the glass transition temperature, and C_1^g and C_2^g are two constants. Typical values of these constants for some polymes are presented in Table 2. As it is quite difficult to determine the viscosity at the glass transition temperature in conventional rheometers [29], thus, the equation for (a_T) should be used having a reference temperature (T_0) different from (T_g), in the form of

$$\ln a_T = -\frac{C_1\,(T - T_0)}{C_2 + (T - T_0)} \tag{22}$$

Table 2: Values of the constants of the WLF equation [24].

Polymers	T_g°C	C_1^g	C_2^g[K]
Polystyrene	100	13.7	50
PMMA	115	32.2	80
Natural rubber	-73	16.8	53.6
SBR	-63	20.3	25.6

Beyond the range of $T_g < T < T_g + 100$, the Arrhenius equation should be used instead of WLF equation [29].

Pressure effect

The dependence of viscosity on pressure alone is a topic that has not been clearly explained because it is difficult to isolate the effect of pressure, and not consider both T and $\dot{\gamma}$ [29]. Even though, there is not so many works demonstrating the effect pressure has in viscosity, many authors assume that viscosity has an exponential dependence on pressure in the form of [44]

$$\eta\,(p) = \eta_0 \exp\left(\chi p\right), \tag{23}$$

where χ is a pressure-dependent coefficient [29], which values may in the range of [10^{-9}, 10^{-8}] Pa^{-1} [42, 45] for molten polymers [29]. Figure 9 illustrates the variation of viscosity with pressure, for different pressure-dependent coefficients.

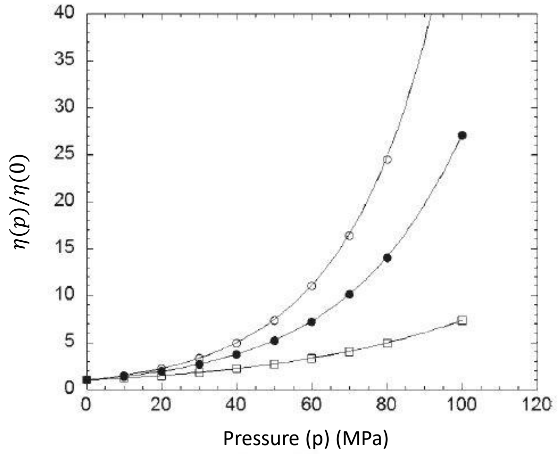

Figure 9: Variation of viscosity with pressure for different values of pressure dependent coeffcient (□ : $\chi = 2 \times 10^{-8}\mathrm{Pa}^{-1}$, ● : $\chi = 3.3 \times 10^{-8}\mathrm{Pa}^{-1}$, ○ : $\chi = 4 \times 10^{-8}\mathrm{Pa}^{-1}$). (Image adapted from [29]).

Finally, combining the models for shear-rate, temperature, and pressure effects on viscosity, we obtain the two main models for GNFs fluid employed in CFD software, the Cross-WLF equation and Cross with Arrhenius dependence. The difference between both is in the application of Arrhenius equation or WLF equation on the determination of the viscosity at a null shear rate (η_0). The Arrhenius equation gives η_0 by

$$\eta_0(T,p) = B\exp\left(\frac{T_b}{T}\right)\exp(\beta p). \tag{24}$$

where B, T_b, β are material parameters [11]. And, the WLF equation gives η_0 by the following equation

$$\eta_0(T,p) = D_1\exp\left(\frac{-C_1(T-T_0)}{C_2+T-T_0}\right), \tag{25}$$

where $T_0 = D_2 + D_3 p$, being D_1, D_2, D_3, C_1, C_2 material parameters, determined experimentally. And finally, viscosity η is given by the Cross model, given by equation

$$\eta(\dot{\gamma},T,p) = \frac{\eta_0(T,p)}{1+\left(\frac{\eta_0(T,p)\dot{\gamma}}{\tau^*}\right)^{n-1}}. \tag{26}$$

2.4. Thermal Properties

During the injection moulding process, three main phenomena related to internal (thermal) energy can occur: heat conduction from the polymer to the mould walls, heat convection from the incoming melt, and shear heating caused by the viscous dissipation developed on the melt. The main properties that govern these phenomena are: specific heat capacity, thermal conductivity, viscosity. The material's viscosity was already discussed in the previous subsections, hereafter the remaining properties will be discussed.

Specific Heat Capacity

Specific heat capacity is the amount of energy that is required to provide to a one unit of mass of a material to raise its temperature by one Kelvin, and conversely, gives a prediction of the amount of energy that is contained within a material at a given temperature [9]. The specific heat capacity is defined as the heat capacity per unit mass of material, the *SI* unit for this property is Joule per kilogram and Kelvin $(J/kg.K)$. It can be measured at constant pressure or with a constant volume, being designated as c_p or c_v, respectively [29].

Table 3 presents the specific heat capacity of some polymers often used in injection moulding process, and of some metals as well. From Table 3 it is possible to conclude that metals need less energy to increase their temperature when compared to polymers [12]. As shown in Figure 10 semi-crystalline polymers and amorphous polymers have similar dependencies on temperature, differing only on the temperature at which discontinuity appears. For amorphous it happens at T_g and, for semi-crystalline polymers it takes place when melt state is reached, at T_m [17].

Table 3: Specific Heat Capacity of some Polymers and Metals [17].

Type	*Material*	*Specific Heat Capacity (J/kg.K)*
Semi-Crystalline	Polyethylene, PE	2260
	Polypropylene, PP	2140
	Polyamide, PA	2180
	Polyethylene terephthalate, PET	1550
Amorphous	Polystyrene, PS	1720
	Polyvinyl chloride, PVC	1220
	Polymethyl methacrylate, PMMA	1800
	Polycarbonate, PC	2093
	Acrylonitrile Butadiene Styrene, ABS	2093
Metals	Steel (AISI 1020)	460
	Steel (AISI P20)	460

In simulations, c_p is usually considered as constant, or temperature dependent[11].

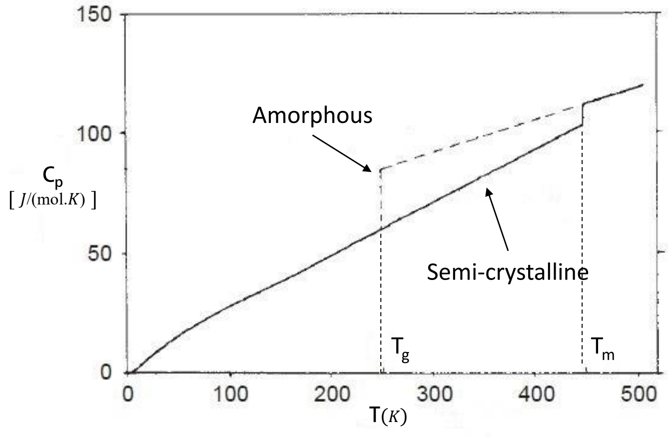

Figure 10: Variation of the specific heat capacity with temperature for different types of polymers. (Image adapted from [17])

Considering a linear interpolation function between two points (T_1, C_{p1}) and (T_2, C_{p2}) for the dependence of heat capacity on temperature, the relation is given by

$$c_p = aT + b, \text{ where } a = \frac{c_{p2} - c_{p1}}{T_2 - T_1} \text{ and } b = \frac{T_2 c_{p1} - T_1 c_{p2}}{T_2 - T_1}. \tag{27}$$

Thermal Conductivity

Polymers present low thermal conductivity when compared to metals, being usually designated by insulators [46]. Table 4 present some values of thermal conductivity for polymers and metals. In Table 4 the different polymeric materials present a thermal conductivity of around $(10^{-1}\,[W/(m.K)])$, while the order of magnitude for metals is $(10^{1}\,[W/(m.K)])$. This means that metals present a thermal conductivity 100 times higher than polymers.

Thermal conductivity has a great influence on the phenomenon of heat conduction. As previously mentioned, in the injection moulding process we have the phenomenon of heat conduction from the melt material to the mould wall, for example. Heat conduction is usually modelled by Fourier's law which is given by

$$\underline{q} = -k\nabla T, \tag{28}$$

where $\underline{q}$ is the heat flux in W/K. k is the thermal conductivity, and T is the temperature in K. Finally, in CFD software, thermal conductivity is usually assumed as constant or considering a linear dependence with temperature given two points $((T_1, k1)$ and $(T_2, k2))$ [11]

$$k = aT + b, \text{where } a = \frac{k_2 - k_1}{T_2 - T_1}, \text{and } b = \frac{T_2 k_1 - T_1 k_2}{T_2 - T_1}. \tag{29}$$

Table 4: Thermal conductivity of some polymers and metals [46].

Type	*Material*	*Thermal Conductivity (W/m.K)*
Semi-Crystalline	High Density Polyethylene, PE	0.44
	Low Density Polyethylene, LDPE	0.30
	Polypropylene, PP	0.12
	Polyamide, PA	0.25
	Polyethylene terephthalate, PET	0.15
Amorphous	Polystyrene, PS	0.14
	Polyvinyl chloride, PVC	0.19
	Polymethyl methacrylate, PMMA	0.21
	Polycarbonate, PC	0.20
	Acrylonitrile Butadiene Styrene, ABS	0.33
Metals	Steel (AISI 1020)	51.9
	Steel (AISI P20)	51.9

2.5. Pressure-Volume-Temperature (PVT) Behaviour

When pressure is applied, polymers shrink and, thus, their specific volume (or its reciprocal, the density) decrease in proportion to the amount of pressure applied [11]. On the other hand, with the increase of temperature, polymeric materials tend to expand, and consequently, their density increases. The relation of density (or specific volume), pressure and temperature is called Pressure-Volume-Temperature behaviour (PVT) [47].

Figure 11 present the evolution of the specific volume with both temperature and pressure for an amorphous and semi-crystalline polymer, respectively.

When at the melt state, both amorphous and semi-crystalline materials have a linear dependence of the specific volume with the temperature. Although, as the pressure increases, the specific volume decreases [11]. When the materials are solids or in temperature range where the solidification happens, the specific volume of amorphous polymers still have a linear dependence with temperature, while, semi-crystalline polymers have an exponential dependence illustrated in Figure 11 [48]. This happens because during crystallization, that takes place as semi-crystalline polymers temperature decrease, the polymer chains arrange themselves to form crystallites and a well organized 3D structure, which results in a much more dense volume and, consequently, in a lower specific volume when compared with the amorphous polymers [11].

In injection moulding both the cavity geometry and solidification of the material

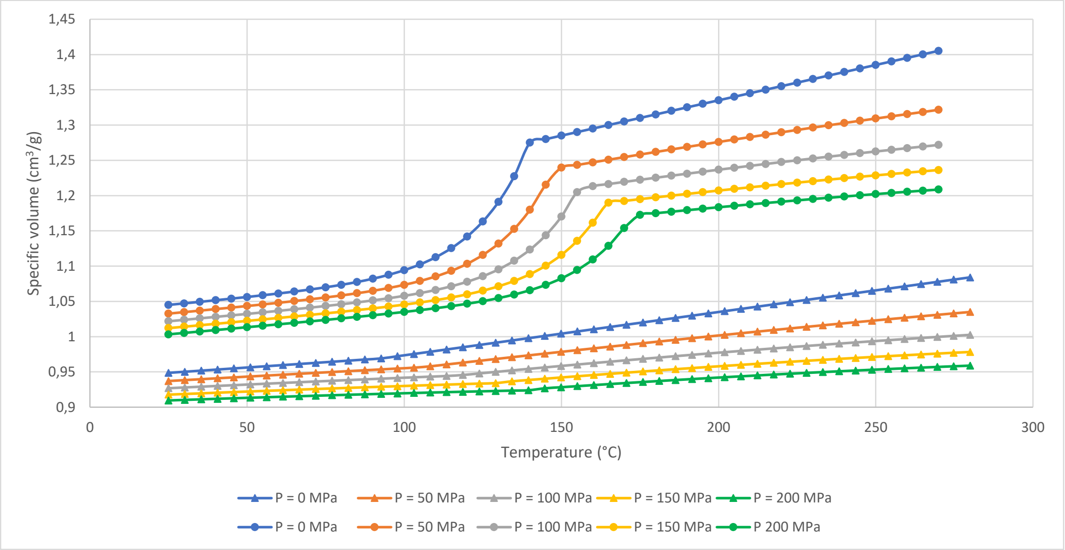

Figure 11: Pressure-Volume-Temperature diagram for an amorphous polymer (ABS) (▲) and for a semi-crystalline polymer (HDPE) (●).

affect the way the material shrinks, and consequently, some defects appear. All materials shrink under load, which means that metals shrink as well under pressure [48]. As metals are more rigid than polymers, the polymer shrinkage will be restricted by the mould and, therefore, thermal stresses might appear, which can produce warpage and distortions on the final product [11]. Additionally, as the walls are at a low temperature when compared to the melt, rapid solidification takes place near the mould walls, which means that the material solidifies, and consequently, restricts the inner polymer shrinkage, which might also contribute to the generation of thermal stresses [11].

Three main models are usually used to model the PVT relation for polymers, they are the constant specific volume model (obviously applied to incompressible materials), Spencer-Gilmore model [17], and the modified Tait model [49]. The first one assumes that the specific volume of polymers is constant, which means that it does not vary with both pressure and temperature. The Spencer-Gilmore [17] is a model derived from the ideal gas law by adding a pressure and temperature correction term to the specific volume [11]. The model has the following form

$$\widehat{V} = \widehat{V_0} + \frac{\widehat{R}T}{p + p_0} \tag{30}$$

where $\widehat{V}$ represents the specific volume of the material, the $\widehat{V_0}$ its reference specific volume, p is the pressure, and T is the temperature of the material. $\widehat{R}$, and p_0 are parameters that need to be experimentally determined [17]. This model has he advantage of being a simple model, although it lacks some accuracy especially when dealing with semi-crystalline polymers

[50, 51]. The most commonly used model to characterize the material PVT behaviour is the modified Tait model [49] because it can deal with PVT relation for both amorphous and semi-crystalline polymers [11], and is given by

$$\widehat{V} = \widehat{V}_0 \left[1 - C \ln\left(1 + \left(\frac{p}{B}\right)\right)\right] + \widehat{V}_t, \tag{31}$$

where

$$\widehat{V}_0 = \begin{cases} b_{1S} + b_{2S}\left(T - b_5\right), & \text{if } T \leq T_c \\ b_{1L} + b_{2L}\left(T - b_5\right), & \text{if } T > T_c \end{cases}, \tag{32}$$

$$B = \begin{cases} b_{3S} \exp\left(-b_{4S}\left(T - b_5\right)\right), & \text{if } T \leq T_c \\ b_{3L} \exp\left(-b_{4L}\left(T - b_5\right)\right), & \text{if } T > T_c \end{cases}, \tag{33}$$

$$\widehat{V}_t = \begin{cases} b_7 \exp\left(b_8\left(T - b_5\right) - b_9 p\right), & \text{if } T \leq T_c \\ 0, & \text{if } T > T_c \end{cases}, \tag{34}$$

where $C = 0.0894$ and the coefficients b_1 to b_9 are coefficients that can be obtained by fitting experimental data. p is the pressure applied to the polymer melt [11]. The indexes S, L, and c, represent solid state, liquid or melt state, and the change/transition respectively.

These models have a particular importance in the definition of the isothermal compressibility coefficient, ι, and in the thermal expansion coefficient, β, since both of them depend on specific volume, and, are given by

$$\iota = -\frac{1}{\hat{V}}\left(\frac{\partial \hat{V}}{\partial p}\right)_T, \tag{35}$$

$$\beta = \frac{1}{\hat{V}}\left(\frac{\partial \hat{V}}{\partial T}\right)_p. \tag{36}$$

3. State of art

In this chapter an overview about the history of numerical modelling applied to the injection moulding process is presented, going from the one dimensional (1D) approaches to the three dimensional (3D) ones. The objective is to give the reader an insight of what has already been achieved in terms of numerical modelling of the injection moulding process, and which fields are still yet to be developed. In the end of this chapter, the same procedure is employed for OpenFOAM®, focusing of the developments made in the computational library for the numerical modelling of the injection moulding process.

Long before the advent of simulation, the injection molding process was already seen as an industrial process with enormous potential [12]. In fact, in the first times, the experience with the observation of the part quality was the key to validate or reject a part. While it was known that the processing conditions affected the final part quality, at that time, it was quite difficult to make a good prediction of the final result, especially because development of morphology was also present, affecting enormously [52]. With the rise of the numerical simulations advent, this triangle of concepts made by the processing conditions, morphology, and final properties started to be investigated, which increased the insights about the process [53]. In fact, while the relation between processing conditions and final properties is well known nowadays [11], the same cannot yet be claimed about the morphology, where work still need to be done to get a full understanding of its physics [54].

The computer-aided-engineering in injection moulding began in the 50's with the works done by Toor et al. [55], where a scheme was developed to predict the average velocity of a polymer melt filling a cold rectangular cavity. These results were used to calculate the time to fill a cavity of a given length. The calculations accounted for heat loss by conduction and used experimentally determined parameters for the effect of both shear rate and temperature on fluid viscosity. During this period, Spencer and Gilmore [56, 57] studied a 1D mould filling using a power-law equation to relate the filling time with the pressure drop.

1970-1979

It was in the 70's when the first mathematical model trying to describe the injection moulding process was proposed [12]. Barrie [58] studied the pressure drop in the feeding system of a disk cavity injection moulded part. Barrie [58] assumed that an elongational viscosity may be needed for the prediction of the cavity pressure near the sprue owing to the elongational deformation rate developed in that location. Ballman and co-workers [59] provided a study of the filling of a thin rectangular cavity applying a non-isothermal model.

During this period, the works done were all focused on simple geometries, which had some academic interest, but it offered little assistance to the industry [12]. The work of Stevenson and co-workers [60] presented the simulation of 1D center-gated disc. In 1977, Nunn and Fenner [61], did a similar study but in a tubular channel.

Additionally, in this decade other works were presented like the research of Kamal and Kenig [62], which analysed the filling, packing and cooling stages. They used the finite difference method (FDM) to deal with temperature and pressure fields. The work of Williams and Lord [63] analysed the runner system using FDM. The work was then adapted to the analysis of the filling cavity with FDM [64].

The 70's have been marked by the creation of the first company dedicated to the injection moulding simulation, which was founded in Australia by Colin Austin in 1978 [12]. The injection moulding simulation software provided by them was based on the works of Kamal and Kenig [62], Lord and Williams [63, 64], Barrie [58], and the work of Austin itself [65]. The company was named *Moldflow*® [66], which after being bought by the *Autodesk, Inc.* in 2008, it still works nowadays under the name of *Autodesk Moldflow*®.

The first years of *Moldflow*® [66] were marked by the fact that their first products were distributed via time-share services due to the prohibitive cost of computers. In terms of products, one of the first products launched by *Moldflow*® was a set of rules to improve the design of both plastic parts and runner systems, named Moldfow Design Principles [65]. In terms of simulation, the early products of Moldflow used the "layflat" technique, which consisted on reducing a 3D problem into a midplane flow [67]. For simple geometries this technique was accurate, but for complex ones, the method was not able to predict accurate results [12].

Considering the simplifications made to the governing equations, these studies were essentially a one-dimensional analysis for the pressure drop on injected moulded parts [67]. In terms of temperature, the software used the same FDM approach used by Barrie [58], with both melt and mould temperatures were considered constant, and the software could already deal with viscous dissipation. Convection and conduction heat transfer were also accounted in the software. Viscosity was modelled as a generalized Newtonian fluid using power-law model presented in the works of Williams and Lord [63, 64]. Due to the simplicity of the "layflat" technique, the simulations were extremely fast which allowed users to use this software to make changes to the geometry of the parts, namely the thickness, and thus improve the design of plastic products [12].

1980-1989

In the 80's injection moulding started to be treated as a multi-stage process, *i.e*, considering not only the filling phase, but also the other phases of the process that need to be optimized [12]. This decade was also marked by the introduction of Finite Element Method (FEM) [12].

During this period, a software named McKam was developed by the university of McGill for the filling and packing stages of injection moulding simulation [68]. McKam used the FDM for the numerical discretization of the governing equations. The software was developed with the objective of predicting other properties like birefrengence and tensile modulus [68]. In 1986, Lafleur and Kamal [69] presented an analysis of the thermo-mechanical history developed during the filling, packing, and cooling stages. Furthermore, he studied the development of crystallinity and orientation in order to estimate the tensile modulus of the injected moulded part. All of these features were predicted using viscoelastic consitutive model [69].

Other papers were published during the 80's, but one of the greatest achievement of this decade came from the Cornell Injection Molding Program (CIMP) group, from the Cornell University created in 1974, with the introduction of the 2D/2.5D analysis of the process [12]. Under the guidance of K. K. Wang and, with the works of Hieber and Shen in 1980 [70] as the base of their developments, they developed a code capable of simulating injection moulding for thin-walled parts. The work of Hieber and Shen was based on the Hele-Shaw equations [71]. This approach considers the Stokes equation on a narrow, planar geometries, where the thickness is infinitesimally small when compared to the width and length [71]. This way, the thickness could be approximated as zero, and the governing equations were simplified into a 2D case [71]. This approach was introduced by Henry Selby Hele-Shaw, and, the method was named after him.

The code devolved by K. K. Wang assumed the Hele-shaw approach, an incompressible flow, and, a symmetric flow field about the cavity center line [72, 73]. They developed an equation to calculate pressure in two dimensions, relating pressure with a property called fluidity introduced by themselves. The temperature equation was based on the same assumptions but solved in 3D.

This method was called 2.5D because pressure field was solved in a 2D, and temperature in 3D. As pressure was not calculated in the thickness, the feature of mid-plane assumption was introduced. FDM was used to solve the temperature field in a 3D way [72, 73]. As the pressure field was known, then an equation for velocity in both length and width directions were derived, and then, pressure, velocity and temperature were able to be calculated. With this approach, other phenomena were able to be predicted like warpage, shrinkage, weld lines, or even morphology development [72, 73]. This work introduced the idea of analysing thin-walled geometries that were much more closer to the injection moulded products than the one-dimensional flow paths analysed hitherto in the "layflat" method and FDM codes [12].

On this time, a group from Eindhoven led by H.E.H Meijer and F.P.T Baaijens, and, with the collaboration of the Philips Centre for Fabrication Technology (Philips CFT) developed codes to simulate injection moulding process. Their first major success was achieved with the works of Sitters [74] and Boshouwers and van der Werf [75], which led to the development of Inject-3, a code that dealt with the filling phase for amorphous materials.

During this decade, in the scientific field, the majority of the works were about the filling phase, there were also some few about the packing, but about the cooling phase there were not many works [12]. As cooling phase was the most time consumption phase of the cycle, some researchers started to investigate this phase in order to increase productivity. Furthermore, the cooling phase had impact on the warpage of plastic products, therefore, it was necessary to account this phase to have a better prediction [76].

The work of Kwon et al. [77] introduced a solution for mould cooling analysis in a 1D approach. Himasekhar et al. [78] developed a 1D conduction problem with FDM for both mould and melt temperatures, which is usually the main phenomena of heat loss in the mould. With their work, Himasekhar concluded that an averaged cycle temperature was accurate enough for mould design purposes, which means that defining an averaged temperature for the mould walls during the whole cycle was sufficient [78]. Later, they introduced a 3D approach where the temperature in the mould was determined using a boundary element method (BEM), similar to what Burton and Rezayat [79] did in their works, and a FDM for the heat transfer in the polymer.

During this time, some works on shrinkage and warpage analysis were published as well [12]. The accurate prediction of warpage relies on a good simulation of filling, packing, and cooling phases, but the works on this decade did not account fot the three phases. While the early work was about predicting residual stress on glass, especially due to the cooling, Isayev [76, 80] considered the residual stress in an amorphous polymers. They demonstrated that the flow-induced stresses tended to be tensile with a maximum value at the surface of a moulded part.

Titomanlio et al. [81] provided the relation between the packing pressures and the development of residual stresses. They proposed a simple model for cooling stress build-up in injection moulding process of a rectangular cavity using a polystyrene [81]. They concluded that teir results were favorably compared to the experimental results presented in literature on that time.

1990-1999

In the early nineties, due to the low computational power of the computers, the works were based on the 2.5D approximation. However, the idea of modelling a 3D geometry with a full 3D analysis were investigated [12].

Santhanam and Wang [82] studied the develpmet of warpage due to difference of temperatures across mould halves. Using both thermo-elastic, and thermo-viscoelastic models, their study provided that both models could predict similar deflections. Chiang et al. [83] introduced models for the packing phase in the early 90's, and were able to predict the pressure history over the entire filling and post-filling stages [83]. They implemented a hybrid FDM/FEM of the generalized Hele-Shaw flow for a compressible viscous fluid, in non-isothermal conditions. The Cross-WLF model was used for the viscosity modelling, thermal conductivity and specific heat capacity were introduced as a function of temperature. For the packing, and post-filling stages it was concluded that the compressibility of the material plays a critical role, and therefore, density cannot be considered constant. All of this works were done using the 2.5D midplane analysis introduced in the 80's.

Other contributions on that time came from the Technical University of Eindhoven, with the works of Douven [84] where he simulated the development of residual stresses through a viscoelastic model for an amorphous polymer. For that, Douven [84] used a compressible Leonov viscoelastic model to predict residual stresses in an amorphous moulded part. The main feature of his work was the implementation of the viscoelastic model in two ways. In the first approach, known as decoupled approach, a generalized Newtonian fluid is imposed for the whole cycle, and, then, the flow kinematics are substituted into the compressible Leonov model. This approach has the assumption that the viscoelastic characteristics of polymers does not affect the flow kinematics. The second approach concerns a fully coupled of the viscoelatic model [84]. The decoupled approach presents a good agreement with the results obtained for the coupled method, meaning that computational resources can be spared, and even arbitrary viscoelastic models can be employed. Authors like Baaijens [85] and Caspers [86] followed the same decoupled procedure to predict the shrinkage, warpage, and elastic recovery of an amorphous moulded part. All this works, once again, were all made in a 2.5D midplane analysis [12].

In Italy, Titomanlio and his colleagues studied the pressure drop on the packing phase using a crystallization model, and concluded that the flow affects the crystallization of the material [87]. This necessity motivated them to do some more works on this feature, and in Titomanlio et al. in 1997 [54] they managed to compute crystallization kinetics as a function of the shear stresses.

Later in this decade, the first 3D code for the filling phase analysis was developed. This first code was described by Hétu et al. [88], using a FEM solution for the pressure, velocity and temperature, as well as the flow front position. The solver employed the Stokes equations. They also used a pseudo-fluid method for the propagation of the flow front that involved the use of a total derivative of a property known as polymer concentration. This property was represented by F and could only have values in $[0,1]$. When $F=0$ the cavity is filled with air, whereas $F=1$ corresponds to a filled region with polymer [88].

Moldflow addopted a search for their 3D solver on this decade, and in the end of the nineties they were able to provide their 3D analysis software based on the work of Friedl [89]. Full 3D filling analysis was introduced to the marketplace by Moldflow in 1998 with the work of Rajupalem et al. [90] and was extended to packing phase in 1999 with the work of Talwar et al. [91].

2000-2020

With the development of three-dimensional solvers, the main fields that still required some work were the post filling related areas: warpage, shrinkage, residual stresses, elastic recoveries, etc. The new century brought some new works on these areas, and in the development of morphology as well, especially in the search for the best relation between processing conditions, morphology development, and final properties [12].

In the early years of the new century, a new method of discretisation started to be applied to injection moulding. Till then, FDM and FEM were the most common approaches in polymer processing. The first one because of its simplicity and efficiency, although it presents some problems in complex geometries The second one, has been adopted to polymer processing due to its flexibility in dealing with arbitrary boundary shapes [92]. However, it produces large sparse matrices, usually with high condition numbers, and, consequently, relies on direct solvers and it needs too much computer space to deal with such sparse matrices [92]. Finite Volume Method (FVM) based on the pressure corrections procedures as the SIMPLE [93] algorithm has been widely employed in CFD applications [94]. The main advantage of the FVM when compared to FEM is that the former is more computationally stable in both computer space and time [92]. In 2001, Chang and Wang [92] introduced the FVM to simulate the IM process, where the pressure correction is made by using the SIMPLE algorithm [93]. On their research, Chang and Wang [92] concluded that both Hele-Shaw and full 3D FVM analysis predicted the same results for thin cavities, with the first one being more efficient. But when simulating thick parts, or parts with abrupt changes of thickness, their 3D analysis was clearly more accurate than the Hele-Shaw model.

Concerning the development of morphology, the several works from Janeschitz-Kriegl and Eder [95, 96] worth be empathized, since they clearly demonstrated the enormous effect of shear on both the crystallization kinetics and the resulting morphology. More precisely, they showed that a short, high shear treatment greatly increases nucleation rate and, hence the crystallization rate, whereas the the morphology is affected by the total shear experienced [97]. Moreover, they concluded also that low shear rates promote the growth of the spherulitic structures, while for higher shear rates promote molecular orientation.

Also in the early stages of this decade, Zuidema [98] conducted the first simulation to predict morphology in injection moulding. He used the Schneider equations [99] to determine the distribution and orientation of spherulitic structures. Still in the field of morphology development, Pantani et al. [100] gave a review of morphology evolution in injection moulded parts, and stated that the current biggest challenge is to link morphology with final properties.

In this new century, and because new techniques derived from injection moulding started to appear, some authors started developing appropriate numerical approaches for these new techniques. In terms of non conventional techniques some works like the one made by Kwon [101] who simulated the injection-compression moulding for thin and large battery housing parts, and the injection moulding of fiber reinforcemnts [102].

OpenFOAM® as Software for polymer processing

Due to the already explained meaning of proprietary software, which prevent small to medium size companies to have access to modelling tools, researchers started looking for free distribution alternatives. Therefore, free softwares for polymer processing started to appear as a solution to the proprietary software. In the area of CFD, OpenFOAM® became a very interesting alternative that allow the simulation of many different flow problems, and developers tried to implement in it modelling tools for the injection moulding process.

OpenFOAM® stands for Open Source Field Operation and Manipulation, and is a free open-source toolbox for computational modelling problems [4]. It has the ability of solving both complex fluid flows and simple flows, involving turbulence, multiphase flows, chemical reactions, etc. It has the particularity of being a free and open-source software. The first means that everyone has free access to it, they do not need to pay licenses like the proprietary software. The second means that the source code is released and the users are allowed to study, change and distribute the software [103]. OpenFOAM uses the FVM to solve governing equations, and includes free tools for meshing, pre and post processing, and it also allows parallelization.

In polymer processing, OpenFOAM has been used for many years already, and in different areas like extrusion, blow moulding, injection moulding, etc. In the extrusion area, many works have been published, for example, in 2016 Habla et al. [104], presented the development and validation of a model to compute the temperature distribution in the calibration stage of extruded profiles. Later, in 2017, Rajkummar et al. [105] presented some design guide lines to support die designer activities, in order to allow the achievement of a balanced flow distribution, even in more complex geometries.

The numerical modelling of injection moulding process in free distribution software is not a novelty, many researchers have been trying to develop it, but not everyone has been completely successful. In 2009, Araújo et al. [1] presented the development of a parallel three-dimensional unstructured non-isothermal flow solver, for the simulation of the injection molding process, with both polymer and air being considered incompressible fluids. Their results presented a good accuracy and reasonable parallel efficiency and scalability.

Making use of the OpenFOAM® computational library, Josefz Nagy [3] introduced a framework description of a compressible two phase fluid model with polymer specific material models for description of filling, packing and cooling phases. Although for simple geometries, he proved that the framework of solving compressible form of continuity, linear momentum, and energy equations with the specific material models for polymers, results in a good agreement between theoretical predictions and experimental observations.

Others researches tried their best in order to succeed in this task, Ana Magalhães [106] in her Master Thesis used the interFoam solver of OpenFOAM®, implemented the energy equation on that solver as well as the Bird-Carreau model, and studied the filling stage, by implementing a boundary condition of letting the air escape the cavity but not the polymer, since air brings many problems to the modelling. She reached the conclusion, that the boundary condition implemented resulted in a higher percentage of filled cavity. She also showed that the difference in the viscosities of polymer and air result in a high gradient of viscosities, and it was one of the main reasons for so many instabilities in the calculation procedure.

Kristjan Krebelj, and Janez Turk [2, 7] implemented both the Cross-WLF model and Modified Tait model in the *compressibleInterFoam* solver, called it *openInjMoldSim*, and applied the code to injection moulding. The objective was to simulate filling and packing phase of the process, considering a compressible flow, and applying appropriate models for polymeric materials to the code. The solver was validated in a 2D demo test case for an amorphous polystyrene grade. More recently, Krebelj et al. published some more papers, concerning this time the stress evolution and ejection analysis [5, 6]. Ana Fontaínhas in her Master Thesis [107], tried to validate the same solver developed by Kristjan with a 3D case study and compared the results with the ones obtained by commercial software *Moldex3D®* [8]. A mesh refinement study was employed, and she reached the conclusion that the results predicted with the OpenFOAM® approximate of the results obtained with Moldex3D®, however she could not state that the results were converged. She also noticed that the open-source solver was 18 times slower than the commercial one. All of this were done considering simple thin rectangular geometries.

Comparing both commercial and open-source programs, the differences are remarkable. The open-source software, especially OpenFOAM® need to have more researchers developing the software, in order to further develop appropriate tools. Although there is some works in filling and packing of the injection moulding process, cooling phase still needs to be further investigated to have a better prediction of all the features of the process. Moreover, at the moment, the majority of the available codes focus specifically in single phases of the process. However, for a good prediction of the injection moulding features, filling, packing and cooling phases need to be model together.

Final remarks

The objective of this state of art was to give an insight of what has already been achieved, and is still need to be done in terms of numerical modelling of injection moulding process. Obviously, not all the work developed so far has been shown, but some examples of the developments made over the history on the different fields of the process were presented. The idea of developing an open-source code for the injection moulding process is not new, but until now, not a single person was able to create a competitive open-source alternative for the proprietary programs that could suit the industrial companies.

4. Numerical Modelling

The governing equations for non-isothermal and compressible viscous fluid flows with a free-surface boundary will be presented in this chapter. The description starts with the general conservation equations (mass, momentum, energy), then the proper constitutive models (presented in the section 2) will be considered and, finally, the equations will be adapted to the injection moulding process. At this phase, the most accepted mathematical models used by the commercial softwares will be presented. Finally, in the second half of this chapter, the way these equations are introduced in numerical modelling will be explained.

4.1. Multiphase flow modelling

4.1.1. General Governing Equations

The fluid flow is governed by three main equations, which are the conservation of mass, the conservation of linear momentum and the conservation of energy. The conservation of mass equation is given by [12], [11]

$$\underbrace{\frac{\partial \rho}{\partial t}}_{\substack{\text{Rate of density} \\ \text{(mass per unit of volume)} \\ \text{variation with time}}} + \underbrace{\nabla \cdot (\rho \underline{u})}_{\substack{\text{Local mass flux} \\ \text{(advection term)}}} = 0, \tag{37}$$

where ρ is the material density, $\underline{u} = (u, v, w)$ represents velocity vector, and t is the time. The conservation of mass equation or, continuity equation [12], states that the rate of change of mass of a fluid element is equal to the balance of mass flow in the fluid element [108]. The conservation of linear momentum equations given by the Cauchy equations has the form of [12]

$$\underbrace{\rho \frac{D\underline{u}}{Dt} = \frac{\partial (\rho \underline{u})}{\partial t} + \nabla \cdot (\rho \underline{u}\underline{u})}_{\text{Inertial forces}} = \underbrace{\nabla \cdot \underline{\underline{\sigma}}}_{\text{Surface forces}} + \underbrace{\rho \underline{g}.}_{\text{Body forces}} \quad , \tag{38}$$

where $\underline{\underline{\sigma}}$ is the stress tensor, and $\underline{g}$ is the gravitational acceleration vector. The expanded version of Equations (38), and the definition of total derivative are given in the Appendix B. The Equations (38) states that the linear momentum rate of change in a fluid particle is equal to the sum of the forces acting on that point [108].

Finally, the conservation of energy equation is given by [12]

$$\underbrace{\rho c_p\left(\frac{\partial T}{\partial t}+\underline{u}\cdot\nabla T\right)}_{\substack{\text{Rate of energy}\\ \text{change of a fluid particle}}} = \underbrace{\beta T\left(\frac{\partial p}{\partial t}+\underline{u}\cdot\nabla p\right)+p\nabla\cdot\underline{u}+}_{\substack{\text{Compression}\\ \text{work term}}} \underbrace{\underline{\underline{\sigma}}:\nabla\underline{u}}_{\substack{\text{Viscous}\\ \text{dissipation}}} + \underbrace{\nabla\cdot(k\nabla T)}_{\substack{\text{Heat}\\ \text{conduction}}} + \underbrace{\dot{Q},}_{\substack{\text{Heat}\\ \text{sources}}} \tag{39}$$

where c_p is the specific heat capacity, T the temperature, p the pressure, β the thermal expansion coefficient, k the thermal conductivity, and, $\dot{Q}$ the heat sources which stands, for example, for the heat generated in the cure reaction of thermosets [12]. The energy conservation principle states that the energy rate of change in each location, is equal to the balance of work done by external forces, viscous dissipation due to the applied shear-rate, to the loss of heat by conduction and to source terms [108].

The only simplification that can be made to the conservation of mass equation will be described later in this section. However, the linear momentum and energy equations are simplified hereafter. Assuming a constant thermal conductivity and specific heat, recalling the constitutive equation for a GNF showed in the Chapter 2, and, neglecting the reaction terms, the following are considered [12]

$$\dot{Q}=0, \tag{40}$$

$$\underline{\underline{\sigma}}=-p\underline{\underline{I}}+2\eta\left(\dot{\gamma},T,p\right)\underline{\underline{D}}, \tag{41}$$

$$\underline{\underline{\sigma}}:\nabla\underline{u}=-p\nabla\cdot\underline{u}+\eta\left(\dot{\gamma},T,p\right)\dot{\gamma}^2. \tag{42}$$

where η the represents viscosity for a GNF fluid, $\dot{\gamma}$ the magnitude of shear-rate given by Equation (13), $\underline{\underline{D}}$ the rate of deformation tensor, and $\underline{\underline{I}}$ the identity matrix. The relation given by Equation (42) is detailed in Appendix C. Finally, the final form of the linear momentum and energy equations, respectively, are simplified as:

$$\frac{\partial\left(\rho\underline{u}\right)}{\partial t}+\nabla\cdot\left(\rho\underline{u}\,\underline{u}\right)=-\nabla p+\nabla\cdot\left(\eta\left(\dot{\gamma},T,p\right)\underline{\underline{D}}\right)+\rho\underline{g}, \tag{43}$$

$$\rho c_p\left(\frac{\partial T}{\partial t}+\underline{u}\cdot\nabla T\right)=\beta T\left(\frac{\partial p}{\partial t}+\underline{u}\cdot\nabla p\right)+\eta\left(\dot{\gamma},T,p\right)\dot{\gamma}^2+k\nabla^2 T. \tag{44}$$

4.1.2. Modelling Injection Moulding

Until now, in this section, the main principles that govern the flow of a non-isothermal and compressible viscous fluid were presented, and simplifications regarding the fluid behaviour were made. Now, the objective is to simplify Eqs. (37), (43), and (44) taking into account the specific features of the injection moulding process. Thus, filling, packing and cooling phases will be described, as well as the governing equation for each one, which are derived from the general equations described above.

Filling Phase

At this phase of the process, the most common approach is to consider an incompressible flow, which means that ρ is constant, (see the works of H. Zhou [11], B. Araújo et al. [1], J. Hétu et al. [109], S. Kim and L. Turng [110], V. Rajupalem et al. [90], L. Xie et al. [20], and D. Cardozo [53]). These authors consider the flow as incompressible because it reduces the number of variables to be computed in every time-step (ρ is constant), and, consequently, the equation of state may disappear, since pressure may be computed from the continuity equation. This means that apart from reducing the computational cost, the simplification does not promote significant errors [29]. Thus, the general mass conservation equation (37) simplifies to

$$\nabla \cdot \underline{u} = 0. \tag{45}$$

However, some authors like C. Fetecau et al. [111], P. Guerrier et al. [112], M. Tutar and A. Karakus [113], and C. Fernandes et al. [114] consider the fluid as compressible in the filling phase, and, therefore, use Equation (37) instead of Equation (45). They use compressibility feature because it gives a better representation of the flow especially near the end of filling, and, becomes even more important if the flow is not balanced [29], which means that flow front does not follow at the same velocity, or if the flow front is divided at some point of the geometry. Furthermore, this gives a better transition for the packing phase where variations of density occur, and therefore, incompressibility cannot be considered [29].

The conservation of linear momentum equation for the filling phase of the IM process is commonly given by Equation (43). Although some authors like H. Zhou [11], J. Liang et al. [115], D. Cardozo [53], R. Chang and W. Yang [92] and V. Rajupalem et al. [90] present this equation, this one is quite generic and they end up making some simplifications as well. Starting from this form, J. Hétu [109] and S. Kim [110] postulate Stokes flow for this phase, in which the inertial forces are considered to be very small when compared with the viscous forces, and thus the linear momentum conservation equation simplifies to

$$-\nabla p + \nabla \cdot \underline{\underline{\tau}} + \rho \underline{g} = 0. \tag{46}$$

Moreover, some authors like, K. Choi and B. Koo [116] and H. Zhou et al. [117], defend that even the body forces can be neglected and then, the last term on the left hand side of Equation (46) is also cancelled.

As the flow is considered incompressible, the compressive work term presented in

Equation (44) can be neglected, meaning that in the filling phase, the conservation of energy equation is given by

$$\rho c_p \left(\frac{\partial T}{\partial t} + \underline{u} \cdot \nabla T \right) = k \nabla^2 T + \eta \left(\dot{\gamma}, T, p \right) \dot{\gamma}^2. \tag{47}$$

If compressibility of the material is considered, then Equation (44) is employed instead. The conservation of energy equation presented above is almost universal through the various works about numerical simulation of the injection moulding process (see the works of like H. Zhou [11], K.Choi et al. [116], C. Fetecau et al. [111], P. Guerrier et al. [112], L. Xie et al. [20], C. Fernandes et al. [114], J. Hétu et al. [109], J. Liang et al. [115], D. Cardozo [53], V. Rajupalem et al. [90]).

Packing Phase

The packing phase is characterized by an almost inexistent flow because this phase takes place when the cavity is almost fully filled [21]. In this phase, material tends to shrink due to the high pressures applied, and, therefore, phenomena like warpage and distortions happen [48]. To compensate and minimize these defects, additional material is injected in mould cavity [11]. Thus, in the packing phase, the main difficulty is to predict and prevent shrinkage of the material which is strongly related to the compressibility properties of the materials, therefore, in this phase incompressibility cannot be considered [12].

Even if flux is almost null, the main Equations ((37), (43), and (44)) are suitable for the packing phase. This means that concerning the conservation of mass equation, Equation (37) is used in this phase as C. Fernandes et al. [114], Huamin Zhou [11] and many others postulate in their works.

Concerning the linear momentum conservation equation, the equations and simplifications made in the filling stage are suitable for this phase as well (see the works of L.Xie et al. [20], C. Fernandes et al. [114], S. Kim et al. [110], and F. Ilinca et al. [118]). Therefore, Equation (43) with the assumptions made in Equation (46) and neglecting the body forces are suitable here too [11].

Finally, the energy conservation equation for the packing phase is given by Equation (44), where the only difference concerning the one used in the filling stage is the including of the compressive work term [11].

Cooling Phase

During the cooling phase, the hot polymer melt needs to be cooled down and will solidify in order to allow the part extraction. During this phase, there is heat transfer within the polymer, from the polymer melt to the mould walls, from mould walls to the cooling channels, and to the air environment [11]. The governing equation for the melt temperature is given by Equation (44), with the difference that flow does not exist anymore, and therefore, it becomes

$$\rho c_p \frac{\partial T}{\partial t} = k \nabla^2 T. \tag{48}$$

For the mould, a schematic representation is presented in Figure 12 which represents the evolution of the mould temperature over the production time.

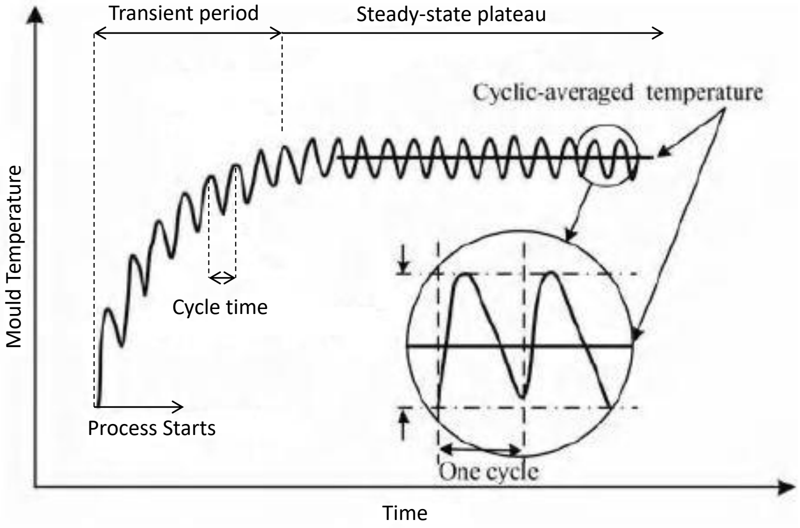

Figure 12: Mould temperature variation with the operating time. Image adapted from [11]

Figure 12 shows that the temperature of the mould in each cycle of injection moulding increases until a steady-state cyclic heat transfer plateau is reached. The mould temperature can therefore be divided into two phases, one cycle averaged temperature, and temperature fluctuation [11] (deviation in relation to the average mould temperature). It is known that the fluctuation part is small compared to the cycle-average component, and, the latter is sufficiently accurate for prediction of temperature distributions [119]. Therefore, the governing equations for the mould heat transfer is divided into a cycle-average mould temperature or performing a full transient analysis [8]. The first is given by

$$k_m \left(\frac{\partial^2 \overline{T}}{\partial x^2} + \frac{\partial^2 \overline{T}}{\partial y^2} + \frac{\partial^2 \overline{T}}{\partial z^2} \right) = 0 \tag{49}$$

where k_m is the mould thermal conductivity, and $\overline{T}$ is the cycle-average mould temperature. In the case of full transient analysis, the mould temperature distribution is obtained by

$$k_m \left(\frac{\partial^2 T_m}{\partial x^2} + \frac{\partial^2 T_m}{\partial y^2} + \frac{\partial^2 T_m}{\partial z^2} \right) = \rho_m c_{p_m} \frac{\partial T_m}{\partial t}. \tag{50}$$

where ρ_m is the density, C_{p_m} is the specific heat capacity of the mould material, and T_m is the mould temperature. The heat transfer from the mould walls to the cooling channels and air environment is accounted applying the correct boundary conditions [8]. Therefore, it is applied the forced convection heat transfer boundary condition, which is based in a heat transfer coefficient [120]. The heat convection is then given by

$$k_m \frac{\partial T}{\partial \underline{n}} = -h_c \left(T_m - T_\infty \right), \tag{51}$$

where $\underline{n}$ is the normal to the surface, h_c is the heat transfer coefficient between the mould and coolant applied. T_∞ is the temperature of the coolant inside the cooling channels [11]. The heat transfer coefficient may be determined as a function of the Nusselt number, or even relating it with the Reynolds and Prandlt numbers [119]. This relation is given by

$$Nu = \frac{h_c D}{k_c} = f\left(Re, Pr\right), \tag{52}$$

where k_c is the thermal conductivity of the coolant, and D is the diameter of the cooling channel. Inside the mould, the melt flow is in a laminar regime, meaning the the flow can be divided into various layers moving at different velocities [11]. However, in the cooling channels, the flow of the coolant is usually in a turbulent regime in order to allow a better heat removing from the mould [120]. Therefore, the Nusselt number can be approximated by the Dittus-Boelter equation [121] that is given by

$$Nu = \frac{h_c D}{k_c} = 0.023 Re^{0.8} Pr^{0.4}. \tag{53}$$

For the heat transfer to the air, the same convection heat transfer gives a good approximation [11].

4.2. Finite Volume Method

Due to their complexity, the governing equations presented on the last section for the IM process cannot be solved by analytical methods. An approach to obtain a solution for them is to apply a numerical method, which transforms the partial diferential equations to a system of algebraic equations, a process commonly called numerical discretization [122]. OpenFOAM® [4] uses FVM in its calculations, so this numerical method was also chosen for this project.

The FVM comprises the following steps:

- Spatial discretization that splits the space domain into a set of control volumes (cells) that when connected fill and bound the space domain again;
- Temporal discretisation that divides the time domain into a finite number of intervals

known as time steps, usually needed for transient problems;

- Equation discretisation which allow the conversion of the governing partial differential equations into a system of algebraic equations, where the quantities of interest (pressure, velocity, etc.) are stored at specific locations of the domain (for the FVM usually in the center of the cells);

In this subsection, we will show how to discretize each term of a general transport equation, and then the calculation procedure (algorithms) commonly used to solve the incompressible and compressible fluid flows are presented (see [123] for a more detailed analysis).

4.2.1. Spatial discretization

In order to compute the solution for a given problem, first the geometric domain is discretised in a computational mesh on which the fields are stored and solved both on space and time.

For the discretisation in time, it is necessary to prescribe the time-step that will be used in calculation, which may be changed during the simulations [122]. The spatial discretisation requires the division of the geometric domain into a finite number of control volumes, commonly called cells, that are connected to each other to form a computational mesh. A typical control volume is presented in Figure 13.

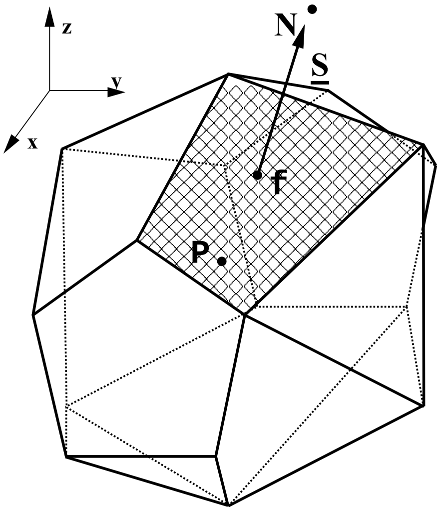

Figure 13: Control volume for the FVM. Where P is the computational point located in the cell centroid, N the neighbour cell or control volume, f the center of the cell face shared by both P and N cells, and $\underline{S}$ is the face area vector that points outwards from the cell, and it is normal to the face f and has the same magnitude of the face area (Image adapted from [123]).

The following section will describe the discretization of a general transport equation in the solution domain.

4.2.2. Discretisation of a general transport equation

Table 5 shows a summary of the governing equations employed in this study. We conclude that the three governing principles have some similarities in their terms [123]. This means that the three governing equations can be approximated by a general transport equation for a field ϕ, where ϕ can represent any scalar.

Table 5: Summary of the governing equations

Conservation equation	Temporal derivarive	Advective term	Diffusion term	Source term
Mass	$\frac{\partial\rho}{\partial t}$	$\nabla\cdot(\rho\underline{u})$		
Linear Momentum	$\frac{\partial(\rho\underline{u})}{\partial t}$	$\nabla\cdot(\rho\underline{u}\;\underline{u})$	$\nabla\cdot\left(\eta\,(\dot{\gamma},T,p)\underline{\underline{D}}\right)$	$\nabla p+\rho\underline{g}$
Energy	$\rho c_p\frac{\partial T}{\partial t}$	$\rho c_p\,(\underline{u}\cdot\nabla T)$	$k\nabla^2 T$	$\eta\,(\dot{\gamma},T,p)\,\dot{\gamma}^2$
General Transport Equation	$\frac{\partial\rho\phi}{\partial t}$	$\nabla\cdot(\rho\underline{u}\phi)$	$\nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right)$	$S_\phi\,(\phi)$

Therefore, the general transport equation presented in Table 5 is given by

$$\frac{\partial\rho\phi}{\partial t}+\nabla\cdot(\rho\underline{u}\phi)-\nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right)=S_\phi\,(\phi) \tag{54}$$

The discretisation practice adopted in the open-source computational library OpenFOAM [4], is second-order accurate in space and time [123]. Equation (54) needs to be satisfied over the control volume V_p of the cell P (see Figure 13). Therefore, integrating Equation (54) we obtain

$$\begin{aligned}\int_t^{t+\Delta t}\left[\frac{\partial}{\partial t}\int_{V_P}\rho\phi dV+\int_{V_P}\nabla\cdot(\rho\underline{u}\phi)\,dV-\int_{V_P}\nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right)dV\right]dt&\\ =\int_t^{t+\Delta t}\left(\int_{V_p}S_\phi\,(\phi)\,dV\right)dt.&\end{aligned} \tag{55}$$

The discretisation of each term present on equation (55) will be treated separately hereafter.

Advective Term

Before introducing the discretization of the spatial terms of the general transport equation it is necessary to introduce the Gauss Divergence theorem that will be used throughout the rest of the chapter. The divergence theorem of gauss relates the divergence of a vector $\underline{a}$ within the volume V_p of the control volume P to the flux of $\underline{a}$ through the surface enclosing the volume [12]. Is is given by

$$\int_{V_p} \nabla \cdot \underline{a}\, dV = \int_S d\underline{S} \cdot \underline{a} = \sum_f \left(\int_f d\underline{S} \cdot \underline{a} \right) = \sum_f \underline{S}.\underline{a}_f, \tag{56}$$

S is the boundary surface enclosing the volume of P, $\underline{S}$ is the face area vector that points outwards from the cell, and f are the various faces enclosing the volume of the cell P. This relation states that the variation of a quantity within the volume of P, is given by the balance of the flow that enters or leaves the cell through its boundary surface.

The discretisation of the advective term is made by applying the Gauss divergence theorem to the volume integral to transform it into a surface integral, and then by application of the mid-point rule to approximate the surface integrals, as folllows [122]

$$\int_{V_p} \nabla \cdot (\rho \underline{u} \phi)\, dV = \int_S (\rho \underline{u} \phi) \cdot dS = \sum_f \underline{S}.(\rho \underline{u} \phi)_f = \sum_f \underline{S}.(\rho \underline{u})_f \phi_f = \sum_f F \phi_f. \tag{57}$$

where F in Equation (57) is the mass flux that passes through the face:

$$F = \underline{S}.(\rho \underline{u})_f. \tag{58}$$

Equations (58) and (57) show that for the calculation of the advective term it is necessary the mass fluxes, and the face value of the variable ϕ calculated from the values in the cell centres of two cells P and N. For the advective term discretization several differencing schemes exist in the literature [108], but here, for the sake of conciseness, we will describe only the Central Differencing (CD) [122] and the Upwind Differencing (UD) [122].

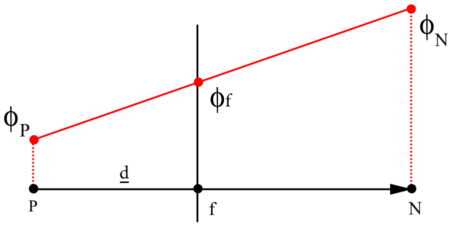

Figure 14: Face interpolation for the CD scheme. (Image adapted from [123])

The CD scheme is illustrated in by Figure 14, where the value of ϕ is linearly interpolated to the face center from the values on the cell centroids P and N, thus ϕ_f is given by

$$\phi_f = f_x \phi_P + (1 - f_x)\, \phi_N, \tag{59}$$

where f_x is the interpolation factor and is given by the ratio of distances between $\overline{fN}$ and $\overline{PN}$ in the form

$$f_x = \frac{\overline{fN}}{\overline{PN}}. \tag{60}$$

This scheme is a second-order accurate scheme as shown by Ferziger and Perić [94]. However, it is not bounded as proved by Jasak [123] and Ferziger [94], which might cause unphysical oscillations for convection-dominated problems [124].

The UD scheme considers on the direction of the mass fluxes or the direction of the flow, and is given by [93]

$$\phi_f = \begin{cases} \phi_P & \text{for} \quad F \geq 0. \\ \phi_N & \text{for} \quad F < 0. \end{cases} \tag{61}$$

Contrarily to the CD scheme, the UD scheme is bounded and allows a better stability during calculation [93], although it is first order accurate [123].

Diffusion Term

Discretisation of the diffusion terms are made in a similar way to the advective term. Therefore, integrating the term in the control volume and converting it into a surface integral we obtain [125]

$$\int_{Vp} \nabla \cdot \left(\rho \Gamma_\phi \nabla \phi\right) dV = \int_s \left(\rho \Gamma_\phi \nabla \phi\right) dS = \sum_f \underline{S} \cdot \left(\rho \Gamma_\phi \nabla \phi\right)_f = \sum_f \left(\rho \Gamma_\phi\right)_f \underline{S} \cdot \left(\nabla \phi\right)_f. \tag{62}$$

Note that the above approximation is valid only if Γ_ϕ is a scalar. On orthogonal meshes, which means vectors $\underline{d}$ and $\underline{S}$ are parallel as shown in Figure 15 , the following expression to calculate the gradient of ϕ on the faces is employed

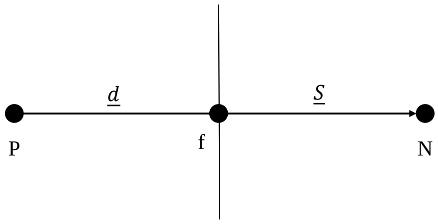

Figure 15: Vectors $\underline{d}$ and $\underline{S}$ in a orthogonal mesh.

$$\underline{S} \cdot \left(\nabla \phi\right)_f = |\underline{S}| \frac{\phi_N - \phi_P}{|\underline{d}|}. \tag{63}$$

which means that the value of ϕ can be calculated from the two cell centers values around the face. Alternatively, the value can be calculated by linear interpolation of the cell-centred gradient for the two cells sharing the face as:

$$(\nabla\phi)_f = f_x (\nabla\phi)_P + (1 - f_x)(\nabla\phi)_N, \tag{64}$$

where

$$(\nabla\phi)_P = \frac{1}{V_P}\sum_f \underline{S}\phi_f. \tag{65}$$

In this last approach, the computational molecule is no longer compact, as the skew and far neighbours are used to calculate the gradient over the neighbouring cell, consequently it is not commonly used for orthogonal meshes [125]. However, for non-orthogonal meshes, which means that $\underline{d}$ and $\underline{S}$ are not parallel anymore as Figure 16 shows (this kind of meshes are usually employed in numerical modelling), it is necessary to compensate the non-orthogonal contribution in the diffusion term [126]. Therefore, the product $\underline{S} \cdot (\nabla\phi)_f$ present in Equation (63) is split into two contributions, being one orthogonal and the other non-orthogonal, given by

$$\underline{S} \cdot (\nabla\phi)_f = \underbrace{\underline{\Delta} \cdot (\nabla\phi)_f}_{orthognal contribution} + \underbrace{\underline{k} \cdot (\nabla\phi)_f}_{non-orthogonal contribution}, \tag{66}$$

and the two vectors $\underline{\Delta}$ and $\underline{k}$ satisfy

$$\underline{S} = \underline{\Delta} + \underline{k} \tag{67}$$

This split allows the orthogonal contribution to be the same as described by Equation (63), limiting the less accurate solution only to the non-orthogonal contribution [127]. In fact, the diffusion term respects the boundness criteria in its differential form, and maintains it if orthogonal meshes are employed. Non-orthogonality may promote unboundness, especially if it is high [94, 123]. Different possibilities for the decomposition of the relation given by Equation (67), such as minimum correction approach, the orthogonal correction approach, and the over-relaxed aproached, which can be assessed in detail in the PhD thesis of Hrvoje Jasak [123].

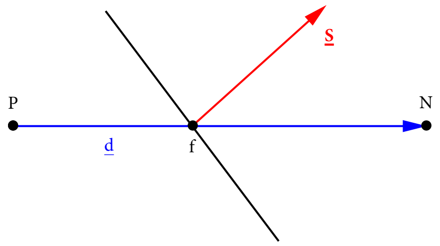

Figure 16: Vectors $\underline{d}$ and $\underline{S}$ in a non-orthogonal mesh. (Image adapted from [123]).

Source Term

The source terms are all the terms that not fit in the convection, diffusion and temporal categories [122]. The source terms S_ϕ can be, in general, functions of ϕ. Usually, a simple way to deal with this terms is to linearise them and then integrate them on the cell volume [122]. The procedure starts with linearisation

$$S_\phi(\phi) = S_u + S_p\phi, \tag{68}$$

where both S_u and S_p may depend on ϕ. Integrating the expression over the control volume we obtain

$$\int_{V_p} S_\phi\left(\phi\right) dV = S_u V_P + S_p V_P \phi_P. \tag{69}$$

More details about this discretisation can be found in the PhD of Hvroje Jasak [123].

Rate of change term

In the latter subsections, the discretisation of spatial terms was presented. Generally speaking, the discretisation transforms surface and volume integrals into discrete sums and expressions that give the face values of the variables as a function of cell values [123].

Recalling general transport equation equation (55) and applying equations (57), (62), (69), the general transport equation can be written in its semi-discretised form as [124]

$$\int_t^{t+\Delta t}\left[\left(\frac{\partial\rho\phi}{\partial t}\right)_P V_P + \sum_f F\phi_f - \sum_f \left(\rho\Gamma_\phi\right)_f \underline{S}\cdot\left(\nabla\phi\right)_f\right] dt = \int_t^{t+\Delta t}\left(S_u V_P + S_p V_P \phi_P\right) dt. \tag{70}$$

This equation is called semi-discretised because it still lacks discretisation of the temporal terms. This process is made by integrating the temporal contributions over the control volume, by assuming a linear variation of ϕ with time, the discretisation of the temporal terms is given by

$$\int_t^{t+\Delta t}\left(\frac{\partial\rho\phi}{\partial t}\right)_P = \frac{\rho_P^n\phi_P^n - \rho_P^0\phi_P^0}{\Delta t}, \tag{71}$$

$$\int_t^{t+\Delta t}\phi\left(t\right) dt = \frac{1}{2}\left(\phi^0 + \phi^n\right)\Delta t, \tag{72}$$

where $\phi^0 = \phi\left(t\right)$ denotes old values from the previous time step, and $\phi^n = \phi\left(t+\Delta t\right)$ stands for the new value at the time step we are solving for. Finally, applying these relations we obtain the so-called Crank-Nicholson form [123]

$$\begin{aligned}\frac{\rho_P\phi_P^n-\rho_P\phi_P^0}{\Delta t}V_P+\frac{1}{2}\sum_f F\phi_f^n-\frac{1}{2}\sum_f\left(\rho\Gamma_\phi\right)_f\underline{S}\cdot(\nabla\phi)_f^n&\\+\frac{1}{2}\sum_f F\phi_f^0-\frac{1}{2}\sum_f\left(\rho\Gamma_\phi\right)_f\underline{S}\cdot(\nabla\phi)_f^0&\\=S_uV_P+\frac{1}{2}S_pV_P\phi_P^n+\frac{1}{2}S_pV_P\phi_P^0.&\end{aligned}\tag{73}$$

From the equation above, it is clear that for the Crank-Nicholson method it is necessary the values of ϕ and $\nabla\phi$, as well as the cell values for old (knowing values) and new time-level (unknowing values). The face values are calculated from the cell values which share the face using the selected discretization schemes. In order to achieve our goal of determining new values for ϕ_P, from equation (73) we obtain a algebraic equation that is assembled for every control volume

$$a_P\phi_P^n+\sum_f a_N\phi_N^n=R_P. \tag{74}$$

where a_P represents the contributions from the control volume of interest P, the a_N represents the contributions from the neighbouring cells, and finally R_P stand for the source terms. This allows to build a system of algebraic equations, which has the form

$$[A]\,[\phi]=[R] \tag{75}$$

Once again, $[A]$ is a sparse matrix matrix composed of the coefficients: a_P, on the diagonal, which includes contribution from temporal derivative, convection, diffusion and the linear part of the source terms [123]; a_N, off the diagonal, are presented the contributions from the neighbouring cells a_N; the $[\phi]$ corresponds to the vector of unknowns, to be evaluated [122]; and $[R]$ respresents all the source terms that not depend on the new values of ϕ or are constant, which includes the old time-level of temporal, convection and diffusion terms, and the constant part of R_P [126].

In fact, the system of algebraic equations represented by Equation (75) allows the resolution of the mass, linear momentum, and energy equations to get the variables dominating the flow. However, from the system of algebraic equations, Equation (75), there will be six variables to be predicted in every time-step (a pressure p, three velocities (u,v,w), density ρ, and temperature T), and there are only five equations to determine them (the mass conservation equation, three linear momentum equations, energy equation) [108]. This means that there is not an explicit equation for determination of pressure, furthermore, the ∇p appears in the momentum equations [128]. This incompatibility along with the non-linear terms present in linear momentum balance equation $(\rho\underline{\underline{u}}^2)$ led to the development of algorithms to overcome these problems, and, allow the calculation of all flow variables [128].

The main algorithms present today in CFD are the SIMPLE, PISO and PIMPLE. The first was developed by Patankar and Spalding [129], and stands for stands for Semi-Implicit Method for Pressure-Linked Equations, and is based on a guess-and-correct procedure for the pressure calculation commonly for steady-state flows [108]. In this method, the time derivatives are not considered (steady-state flows), and there is a correction term to pressure that is not consider also, which led Moukalled et al. [126] to consider it not consistency, and therefore, it needs relaxation factors to achieve the desired stability and convergence rate [128].

The PISO algorithm which stands for Pressure Implicit with Splitting of Operators, developed by Issa [130], was originally introduced in the computation of unsteady compressible flows. The main differences between this method and SIMPLE are the inclusion of the time derivative terms and the consistency of the pressure-velocity coupling equation [128]. This way, using this method, the relaxation factors are not required, although it always should respect a convergence criteria that is the Courant number should never be higher than one, to guarantee that the information is passed from one cell to the adjacent cell and not to two or three cells upfront [94].

Finally, PIMPLE algorithm combines the properties of both SIMPLE and PISO, *i.e*, it presents consistency (PISO characteristics), along with relaxation factors (SIMPLE characteristics), with the advantage of allowing courant numbers much higher than one, and therefore, drastically increasing the time-step [128]. In each time-step, applying the relaxation factors and iterating as many times as needed to guarantee that the explicit parts of the equations are converged, the convergence rate is increased, and then we go on in time. This way, the Courant number can be higher than one increasing the time-step of the numerical calculation [128].

More details about SIMPLE algorithm can be found on Moukalled et al. [126], for PISO algorithm on Ferziger and Perić [94], and for PIMPLE algorithm on Holzmann [128] book.

Multiphase flow

To end the chapter of numerical modelling, an approach to track the portion of the different phases inside the cavity that accounts for both phases (polymer and air) is required. There are different methods for this feature, however, in this thesis we present the one that is employed in the *openInjMoldSim* solver, and in the proprietary software Moldex3D® that is the Volume of Fluid method (VOF) [131]. VOF is a volume method that is based on an indicator function that represents the material location inside the cavity with time [122]. The indicator function is bounded between $[0,1]$.

The VOF approach for compressible fluids follows the transport equation given by

$$\underbrace{\frac{\partial \alpha}{\partial t}}_{\substack{\text{Rate of}\\ \text{change of } \alpha}} + \underbrace{\nabla \cdot (\alpha \underline{u})}_{\substack{\text{balance of } \alpha \text{ flux}\\ \text{through the faces}}} + \underbrace{\nabla \cdot [\underline{u}_r (1-\alpha)\, \alpha]}_{\substack{\text{artificial velocity term}\\ \text{for interface compression}}} = \underbrace{S_p + S_u}_{\text{Source terms}} \quad , \qquad (76)$$

where α represents the fraction volume of polymer inside the computational cell. When $\alpha = 1$ the cell is full of polymer, and when $\alpha = 0$ is full of air, as stated in Figure 17. Obviously, the melt front stands for values between both $0,1$. S_p and S_u represents two terms that account for material's compressibility [122], thus, are only presented in the presence of compressible formulations. The artificial interface compression term is an extra term that only happens when $0 < \alpha < 1$, meaning at the flow front, and it is perpendicular to the interface, allowing therefore, its compression [122]. In Figure 17 it is shown the effect of the artificial interface compression term. The abbreviation named "IC", means interface compression, which means that the artifical term of Equation (76) is presented, and, the abbreviation "NIC" stands for the behaviour of the interface without the compression term.

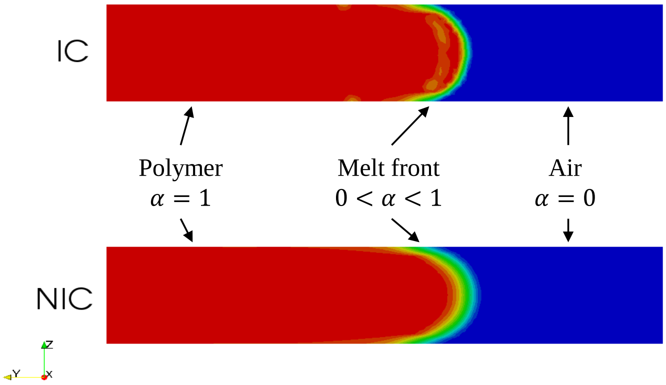

Figure 17: Volume of fluid: α distribution in an injection moulding case study.

Figure 17 shows that when the artificial interface compression term is not presented, the flow front becomes much more diffusive, resulting in a larger flow front.

As the injection moulding process requires two phases inside the cavity (polymer and air), with this method the calculation domain becomes a one-phase system, in which properties are averaged between the values of the polymer and air, weighted with the phase indicator (α) value [107].For example,

$$\rho = \alpha\rho_{Poly} + (1-\alpha)\,\rho_{Air}, \tag{77}$$

$$k = \alpha k_{Poly} + (1-\alpha)\,k_{Air}, \tag{78}$$

$$c_p = \alpha c_{p_{Poly}} + (1-\alpha)\,c_{p_{Air}}, \tag{79}$$

$$\eta = \alpha\eta_{poly} + (1-\alpha)\,\eta_{air}. \tag{80}$$

where the subscript "poly" and "air" represents the polymer and the air phases, respectively.

5. Case studies

In this chapter, first a description of the solvers and of the assessment procedures employed in the different case studies is presented and, then, the case studies employed in this work are introduced. The first Case Study is a simple introductory case for the verification of the numerically computed velocity and pressure fields in a mesh refinement study using a Newtonian model. The second case study is employed to obtain more insights about the process, using the exact same solver, boundary conditions, etc, two variations of the equation of state (compressible and incompressible) are compared in terms of accuracy and performance. Finally, in the last case study, a comparison between the results obtained with the open-source solver and a proprietary software widely employed in the industry is made concerning accuracy and performance of both softwares.

5.1. Solvers and Assessment Methods

In this sub-section the solvers employed in the case studies are presented, and the procedures for the assessment of the different results are explained.

5.1.1. Solvers

In the different case studies presented in this work, two main softwares were used, one open-source and one proprietary. The open-source software employed was OpenFOAM®, more specifically, *openInjMoldSim* solver developed for the simulation of the injection moulding process [5]. The proprietary software was Moldex3D® [8], which is a software widely disseminated in the industry.

The *openInjMoldSim* formulation for the filling phase studied in this work presents the mass (Equation (37)), linear momentum (Equation (43)), and the energy conservation equations (Equation (44)). As rheological models, the solver presents the Newtonian model (Equation (8)) and Cross-WLF (Equations (26) and (25)) as an inelastic model for the constitutive equation. For the equation of state, the solver uses the Modified Tait model (Equation (31)). As referred before, the solver utilizes the VOF method (Equation (76)) to track the phases distribution evolution.

In the results obtained in this Master thesis, two variations of this solver where used. One of them, is the complete solver, with the equations and methods explained in the last paragraph, which is called "*openInjMoldSim*". The second variation has the difference of not considering the artificial interface compression of Equation (76), which is named "*noInterfaceCompression* (NIC)".

Moldex3D® is a very complete software for the simulation of injection moulding process. In terms of governing equations, this commercial software considers mass conservation equation for compressible flows (Equation (37)) and for incompressible flows (Equation (45)). For the linear momentum equation it considers Equation (43) just like the open-source solver, but has the capability of using the Stokes flow given by Equation (46). For the energy equation, the proprietary software does not consider the compression work term done by the compressibility characteristics of the material, which means that the energy equation is given by Equation (47). For constitutive models, Moldex3D® presents Cross-WLF (Equations (26) and (25)), just like *openInjMoldSim*, and Cross-Arrhenius dependence given by Equations (24) and (26). For the equation of state, Moldex3D® presents the incompressible formulation, Spencer-Gilmore (Equation (30)) and modified Tait model (Equation (31)). For the flow front, the proprietary software uses VOF just like the open-source solver.

5.1.2. Verification/Assessment Methods

In the analysis of the case studies presented in this work, some methods used were common to two or even to all of them. Therefore, for organization purposes the methods for assessment and verification used are presented here.

Switch-over time

In different case studies, the time that takes the material to reach the volume-pressure switch-over point will be investigated. This marks the end of the filling phase and the beginning of the packing/holding phase of the process. Therefore, at the switch-over point, instead of forcing a flow rate, a switch is made, and pressure is imposed, as explained in Chapter 2. This makes the switch-over point an important instant for the verification and analysis of the different fields. Assuming incompressibility, the equation that gives the time that takes the material to reach the switch-over point, t_{SO}, is given by

$$t_{SO} = \frac{V}{A \times U}, \tag{81}$$

where U represents the initial velocity at the inlet face, A the area of the inlet face and V is the volume for the switch-over point. The switch-over point was defined in all case studies as 98% of cavity volume filled.

Average Properties

For the assessment of the open-source solver and commercial software, throughout the three case studies, minimum, average, and maximum properties will be calculated. Considering a number of cells n, the average values are weighted by the cells in the form

$$\overline{\phi} = \sum_{i=1}^{n} \frac{|C_i| \times \phi_i}{|C_i|}, \tag{82}$$

where $|C_i|$ can be either the area or volume of each cell. When $|C_i|$ is constant, the Equation (82) becomes the arithmetic mean. ϕ_i is the property of interest in each cell.

Richardson's extrapolation

The Richrdson's extrapolation is the best method for discretization error estimation [132]. It is a method that can be applied to any discretization procedure either differential and integral equations such as FVM [133]. This method is used in this work to extrapolate the values of pressure, velocity and temperature, and estimate the error evolution with mesh refinement. Richardson's extrapolation is usually applied when the values of a property ϕ are known for three meshes with the same grid refinement factor ratio between them [132]. Therefore, the Richardson's extrapolation is given by

$$o = \frac{\ln|(\phi_3 - \phi_2)/(\phi_2 - \phi_1)|}{\ln R} \tag{83}$$

$$\phi_{ext} = \frac{R^o\phi_2 - \phi_3}{R^o - 1}, \tag{84}$$

And the error is estimated by

$$e_i(\%) = \frac{\phi_i - \phi_{ext}}{\phi_{ext}} * 100 \tag{85}$$

where ϕ_{ext} is the variable of interest that we want to extrapolate, indexes $3,2,1$ represent the value of the variable for the three levels of refinement, o is the apparent order, given by Equation (83). R is the grid refinement factor that should be greater than 1.3, and in all case studies we assumed the value of 2, because the number of cells in each direction is doubled for consecutive mesh refinement levels [134].

This relation of the Richardson's extrapolation lays on the assumption that there is not an oscillatory behaviour of the variables to extrapolate, meaning that it presents a monotonic and assymptotic evolution with the mesh refinement [134]. However, many times, this does not happen, and oscillatory behaviours are observed in the variables [132]. In those cases, instead of using Equation (84), a different variation of the method can be used. Knowing the order of accuracy of the scheme used in the calculation procedure, instead of using three meshes, the extrapolated value can be obtained using only the two more refined meshes [135], by

$$\phi_{ext} = \frac{R^0\phi_4 - \phi_3}{R^o - 1} \tag{86}$$

where the only difference concerning Equation (84) is that the apparent order (p) is 1 for schemes of first level order of accuracy, or 2 for second order [135].

5.2. Case Study 1: Filling of a cylindrical cavity with a Newtonian fluid

This first Case Study covers the flow of a Newtonian fluid along a cylindrical channel, and is used to verify if the numerically developed code is well implemented. For that purpose, velocity profiles obtained from the numerical calculations are compared to the analitycal, since for the constitutive model used, the Newtonian fluid, there is an analytical solution for both fields proposed by Liang [115]. Furthermore, by applying two variations of the modified Tait model for the equation of state, some insights about the effects of compressibility will be evaluated. Additionally, we performed a mesh refinement study to obtain converged results with sufficient accuracy, measured by the difference between the obtained value in a specific mesh and the extrapolated one, by using Richardson's extrapolation technique [132].

5.2.1. Geometry and boundary patches

The geometry of the cavity employed in this first Case Study is a cylindrical channel illustrated in Figure 18. It has a length of 100 mm and a diameter of 20 mm (the 2D technical drawing is presented in the Appendix D).

The boundary patches at the faces defining the geometry are also shown in Figure 18. The material enters through the inlet boundary, and leaves by the outlet, the remaining faces are impermeable walls, which only experience heat flux between the polymer and the mould walls.

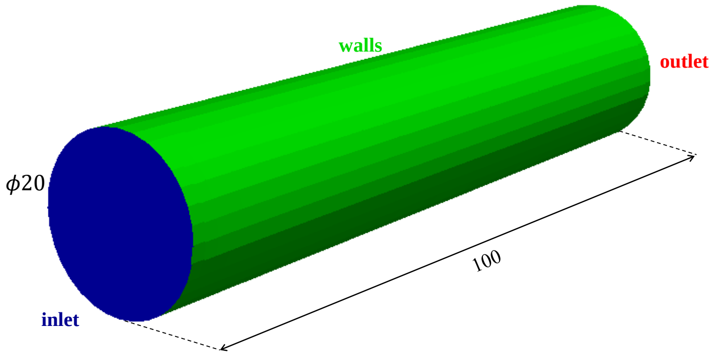

Figure 18: Cavity geometry and boundary patches for Case Study 1 (Dimensions in mm).

5.2.2. Meshes

Three meshes were employed with different degrees of refinement for mesh sensitivity study purposes, a coarse (M1), a medium (M2), and a refined (M3). The number of cells of the coarsest mesh are doubled in each direction to obtain (M2), and the same procedure is applied to (M2) to obtain (M3). Table 6 shows the total number of cells, and the approximate number of cells along the cavity diameter. The OpenFOAM® utility *cfMesh* [136], was used to generate the meshes, more specifically, the Cartesian mesh application is employed to produce a majority of hexaedrical cells.

Table 6: Mesh refinement study parameters used in Case Study 1.

Meshes	*Number of cells*	*Cells along diameter direction*
M1	4 368	10
M2	34 280	20
M3	263 016	40

5.2.3. Material properties

As mentioned before, the fluid employed in this case study is assumed to be Newtonian, therefore the Newtonian model implemented in the open-source solver developed by Kristjan Krebelj [5] was employed. The material used in the simulations was analogous to the General Purpose Polystyrene (GPPS) - Styron 678, from Americas Styrenics, which is widely employed in the injection moulding process, and can be in the future used on experimental studies, because it is available at the Department of Polymer Engineering from the University of Minho. The values of viscosity (μ), the specific heat capacity (C_p) and thermal conductivity (k) for both polymer and air are presented in Table 7.

Table 7: Physical properties of both air and polymer considered for the Case Study 1.

Phase	μ *[Pa.s]*	c_p *[J/(kg K)]*	*k [W/(m K)]*
Polymer	310.0	2100	0.1500
Air	0.1	1007	0.0263

The viscosity of the air is higher than the theoretical value of 1.84^{-5} (Pa.s), because the gradient of viscosities using the latter is too high (magnitude of 10^7 Pa.s) and cause instabilities in the numerical procedure. Therefore, the value of 0.1 Pa.s is used to obtain a better convergence ratio without affecting the final results [5]. The PVT behaviour of the material was modelled using two variations of the modified Tait model as equation of state, both illustrated in Figure 19.

The compressible variation presented in Figure 19 shows that the specific volume of the material decreases with pressure, and, increases with temperature following a slope until T_g (glass transition temperature) is reached. After T_g, the slope increases as is usually

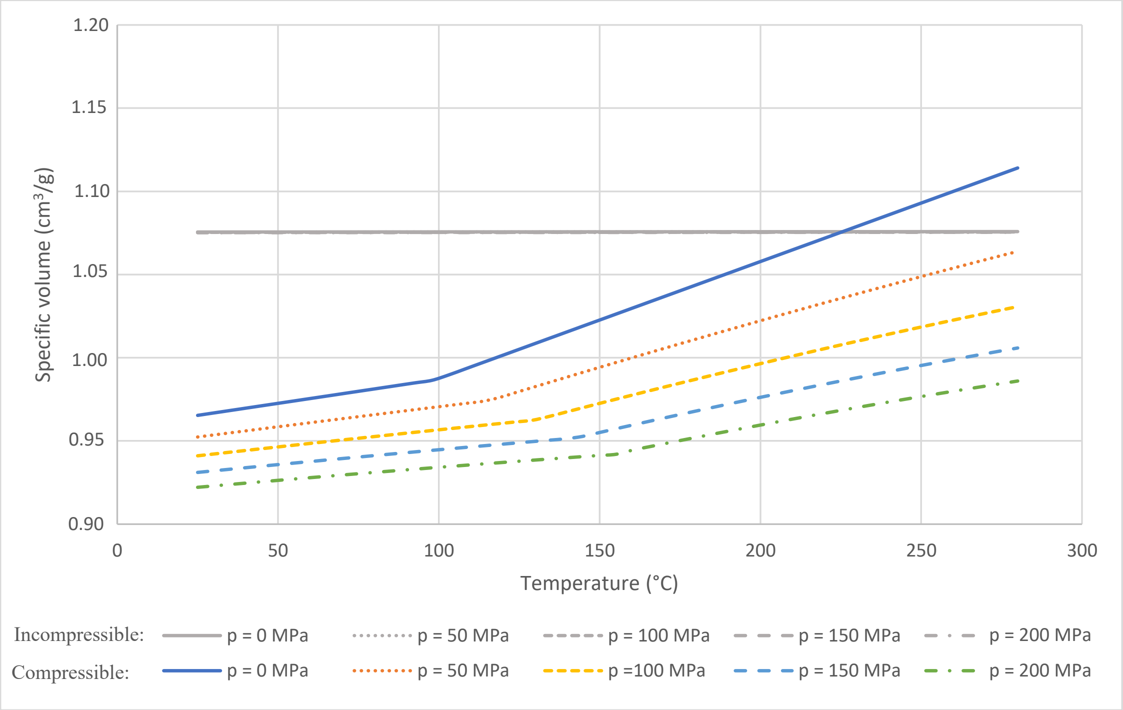

Figure 19: Variation of the polymer specific volume with temperature and pressure for the two variations of the Modified Tait model. Coloured lines - compressible, Gray lines (all overlapped) - incompressible.

observed in amorphous polymers. The incompressible variation curves are all overlapped, meaning that the specific volume, and consequently, density, does not vary. With this second variation, we mimic the truly incompressible formulation for polymers. The coefficients of the modified Tait model for each variation presented are shown in Table 28 in Appendix E.

5.2.4. Initial and boundary conditions

The initial and boundary conditions describe the state of the polymer at the beginning ($t = 0$ s) of the flow, and materials behaviour at the patches. The initial and boundary values used in this case study are presented in Table 8.

For the temperature field, at the inlet it was assumed that the material entered the cavity with a constant temperature of $T = 230°C$. At the walls, a convection heat flux boundary condition between the polymer melt and the mould walls was considered, by using the *externalWallHeatFluxTemperature* boundary condition available in OpenFOAM [4]. For this boundary condition, we have the mould wall temperature equal to $T_{mould} = 50°C$ and the heat transfer coefficient, equal to $h = 1250W/\left(m^2K\right)$ [5]. At the outlet boundary, a null normal gradient was employed, which imposed a fully developed temperature profile, by setting the outlet face temperature equal to the cell centre temperature, from the boundary adjacent cell [107], which is achieved in OpenFOAM by

Table 8: Initial and boundary conditions for Case Study 1.

	Temperature (T)	**Velocity (U)**	**Pressure (p)**	**phase (α)**
Inlet	230 °C	4 m / s	Null normal gradient	1
Outlet	Null normal gradient	Null normal gradient	10^5 Pa	Null normal gradient
Walls	Heat convection T_{mould} = 50 °C h = 1250 (W/(m^2 K))	0 m / s	Null normal gradient	Null normal gradient
Initial Condition	230 °C	0 m / s	10^5 Pa	0

imposing the *zeroGradient* boundary condition [137]. The initial condition for the temperature inside the cavity (air) was considered to have the same value of the inlet melt temperature.

The velocity field is initialized in all cells with U = 0 m/s. At the inlet, the velocity imposed is equal to U = 4 m/s. At the mould walls, the commonly used no-slip (null velocity) boundary condition was employed. Similarly to the temperature field, at the outlet a null normal gradient boundary condition was considered for velocity.

For the pressure field we have considered the atmospheric pressure 10^5 Pa, as the initial condition and at the outlet face. At the inlet, and at the mould walls as velocity is already imposed, pressure cannot be imposed [3], thus, null normal gradient boundary condition was applied.

The phase indicator function is initialized in all the domain cells as $\alpha = 0$ because the cavity only has air inside at the initial state. The melt enters by inlet, therefore, $\alpha = 1$ at the inlet face. Finally, at the mould walls and outlet the null normal gradient boundary condition was assumed, meaning that they will have the value of the cell centre containing the boundary face.

5.3. Case Study 2: Effects of compressibility on the filling of a rectangular cavity with a *Cross-WLF* fluid

In this second Case Study the effects for different variations of the equation of state for the PVT behaviour of the polymer were evaluated. The equation of state gives the variation of the specific volume with both temperature and pressure fields, and, therefore, allows to account for the material's compressibility. In this study, taking into account the already long discussion among several authors regarding the compressibility of polymeric materials in the filling phase of the injection moulding process, two variations of the modified Tait model for the equation of state (one compressible and the other incompressible) were tested. The objective is to compare both in terms of performance and accuracy, and understand if the incompressible formulations can be used in the filling stage of the injection moulding process without loss of the accuracy of the results.

5.3.1. Geometry and boundary patches

This case study describes the filling of a rectangular cavity with a cylindrical insert, illustrated in Figures 20 and 21. As shown in Figure 20, the cavity has a constant thickness of 4 mm, a width of 40 mm, and a length of 150 mm. The cylindrical insert has a diameter of 15 mm, and its center is located at 55 mm in length. The 2D technical drawing of the part is presented in Appendix F. In terms of flow behaviour, this case study presents the flow front separation due to the cylindrical insert, and their joining after the passage through the insert. This behaviour may bring problems in the final properties of the plastic products, however, it is usually presented in the injection moulding process [15]. Figure 21 shows four instants of the flow path for this case of study where the separation and the junction of the flow front can be seen, where $t_{SO} = 0.98$ seconds.

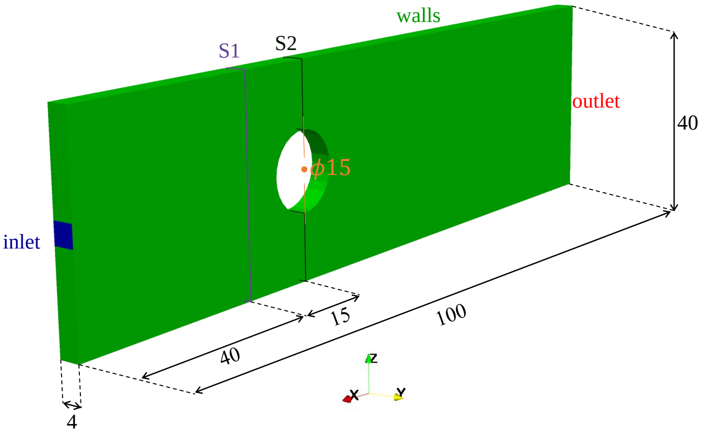

Figure 20: Geometry and Boundary patches for Case Study 2 (Dimensions in mm).

The boundary patches used in this Case Study are the same used for Case Study 1 (Section 5.2), and they are presented in Figure 20.

5.3.2. Meshes

The mesh refinement strategy employed in this Case Study is the same employed in Case Study 1 (Section 5.2), meaning that three degrees of refinement are employed, one coarse (M1), one medium (M2), and one refined (M3). Table 9 shows mesh parameters for

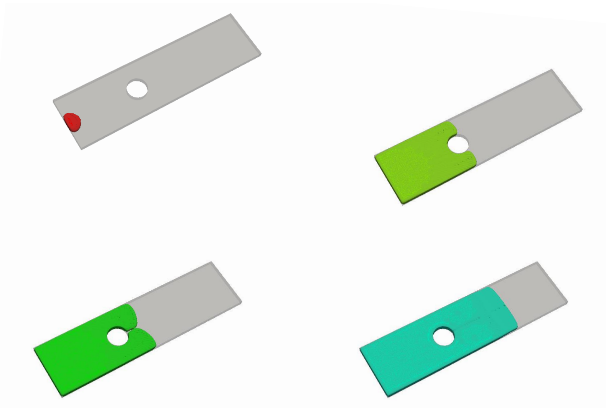

Figure 21: Melt front location for Case Study 2 at t/t_{SO}: $0.02, 0.39, 0.49$, and 0.71.

each level of refinement in this second Case Study. All meshes were obtained using the *cfMesh* [136] utility available in OpenFOAM®, applying a Cartesian mesh.

Table 9: Mesh refinement study data used in Case Study 2.

Meshes	*Number of cells*	*Cells along thickness direction*
M1	23 712	4
M2	187 480	8
M3	1 494 064	16

Table 9 shows that the number of cells along the thickness direction is doubled in each level of refinement, making the total number of cells circa 8 times higher for each level (due to the same procedure applied for the other two directions).

5.3.3. Materials properties

Polymer melt will be modelled as a generalized Newtonian fluid, by using an inelastic model to approximate the variation of its viscosity with shear-rate, temperature and pressure. The material used in this case study is the GPPS Styron 678, from Americas styrenics. The inelastic model chosen to approximate the material's viscosity was the *Cross-WLF* model, commonly used on the commercial softwares, and its coefficients are shown in Appendix E. The evolution of the polymer viscosity with shear-rate and temperature is presented on Figure 22. As stated in Chapter 2, Figure 22 shows that the viscosity of the polymeric material decreases with the increase of the shear-rate, and with the increase of temperature.

The main feature of this case study is the equation of state, *i.e*, the modelling of the material Pressure-Volume-Temperature behaviour. Thus, taking the Modified Tait equation as base model for equation of state, the same compressible and incompressible variations

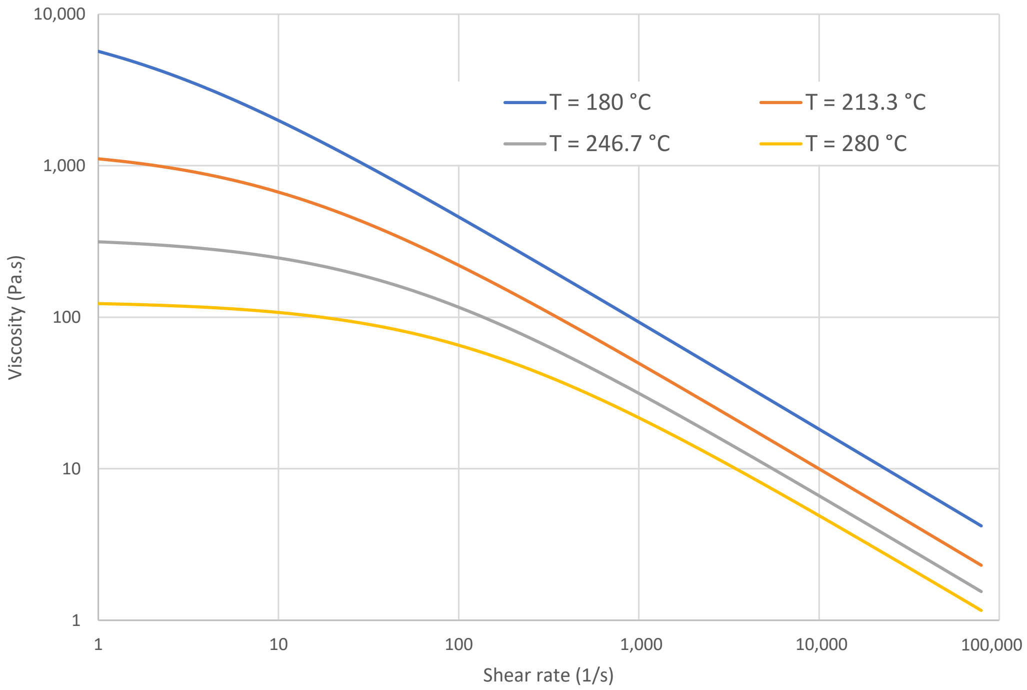

Figure 22: Variation of the viscosity with shear-rate and temperature for the Styron 678, from Americas Styrenics.

employed in Case Study 1 (Section 5.2.3) were simulated. Therefore, Figure 19 presents both variations employed. In this Case Study, the compressible one will be assumed as the reference variation because it is the most realistic, and the differences promoted by the incompressible variation will be measured. The properties for the air are the same of those used in Case Study 1 Section 5.2, (Table 7).

5.3.4. Initial and boundary conditions

The initial and boundary conditions for this Case Study are very similar to the ones used in the Case Study 1, (Section 5.2), presented in Table 8 with the major difference being the imposed injection velocity, which is now defined as U=1.46 m/s. The initial and boundary conditions for this second Case Study are presented in Table 10.

5.4. Case Study 3: Filling of a tensile test specimen

This third Case Study describes the simulation of the filling stage of a tensile test specimen. The aim of this case study is to compare the results obtained with the open-source software *OpenFOAM*® and the commercial software *Moldex3D*®. The accuracy and performance of both softwares will be computed for different meshes. Finally, as stated in Chapter 1, the lack of successful case studies is still one major drawback for the non dissemination of open-source codes. Therefore, this third Case Study was employed in order to fill that gap. For that, a part which could be produced in the Department of Polymer

Table 10: Initial and boundary conditions for Case Study 2.

	Temperature (T)	**Velocity (U)**	**Pressure (p)**	**phase (α)**
Inlet	230 °C	1.46 m / s	Null normal gradient	1
Outlet	Null normal gradient	Null normal gradient	10^5 Pa	Null normal gradient
Walls	Heat convection T_{mould} = 50 °C h = 1250 (W/(m^2 K))	0 m / s	Null normal gradient	Null normal gradient
Initial Condition	230 °C	0 m / s	10^5 Pa	0

Engineering of University of Minho was chosen. It is a simple geometry with a low thickness commonly used for material mechanical properies characterization [115, 138], and, with the feeding system included in the analysis, more results can be found, like the evolution of the pressure throughout the plastic product and feeding system.

5.4.1. Geometry and boundary patches

The tensile test specimen mould cavity geometry is shown in Figure 23. The sprue is represented by the first conical channel and the cold slug. The melted material then flows to the main runner and, after, enters in the secondary runner that changes the flow melt direction. Before reaching the cavity, melt passes through a very thin and small channel called gate, which will cause a pressure peak [11]. Finally, the melt fills the mould cavity. The part has a constant thickness of 4mm, typical of an injected moulded part, and its main dimensions are presented in Figure 23. The 2D drawing of both part and feeding system is presented in Appendix G and Appendix H, respectively.

The boundary patches employed in this Case Study are similar to the ones of the previous case studies, and are presented in Figure 23 as well. There is an inlet face from where the melted material enters, and an outlet face from where air exists, the remaining faces are impermeable walls that only experience heat flux between the mould walls and polymer melt. Figure 23 shows also a slice (S1) from where some contours will be extracted.

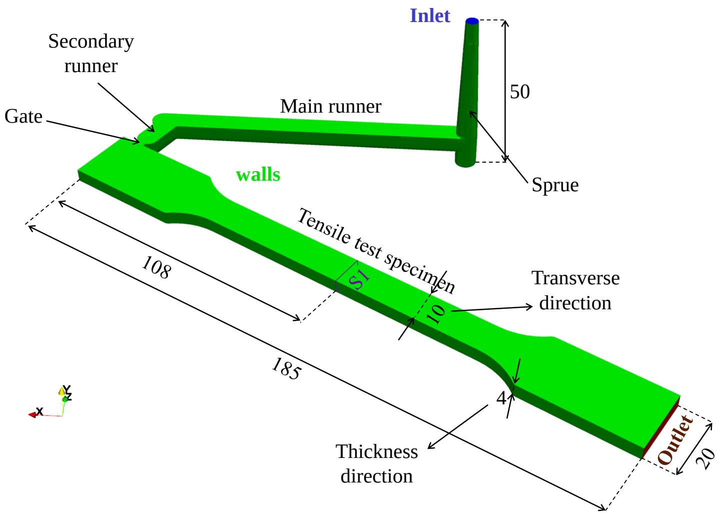

Figure 23: Geometry and patches for the tensile test specimen Case Study (Dimensions in mm).

5.4.2. Meshes

Since the objective of this study is to make a comparison between OpenFOAM®, and Moldex3D® softwares, the simulations to be performed should be made in the same meshes. However, as Moldex3D® only accepts meshes made in its workbenches, and OpenFOAM® could not read those meshes, the simulations could not be made exactly in the same meshes. Therefore, the approach employed was to make simulations in meshes with similar number of cells, until reaching converged results. The same procedure employed in the previous case studies was also employed here, thus, Tables 11 and 12 describe the meshes used in OpenFOAM® and Moldex3D®, respectively. The OpenFOAM® utility *cfMesh* [136], was used to generate the meshes, more specifically, the Cartesian mesh application is employed to generate a majority of hexaedrical cells. The Moldex3D® meshes were created using the program workbench Moldex3D R16 Designer [8], applying a boundary layer mesh (BLM).

Table 11: OpenFOAM® meshes employed in Case Study 3.

OpenFOAM®			
Meshes	*Number of cells*	*Cells along transverse direction*	*Cells along thickness direction*
M1	30 091	8	5
M2	272 149	18	11
M3	2 110 987	37	21

Tables 11 and 12 show that the number of cells for each level of refinement is very close between both softwares, which was a concern with the mesh refinement study. Tables 11 and 12 show that the number of cells for each direction of the tensile test specimen is

Table 12: Moldex3D® meshes employed in Case Study 3.

Modex3D®			
Meshes	*Number of elements*	*Cells along transverse direction*	*Cells along thickness direction*
M1	29 625	8	2
M2	272 409	12	5
M3	2 101 139	26	10
M4	15 304 010	54	21

doubled in consecutive mesh refinement levels, and, consequently, the total number of cells is 8 times higher than in the previous level. Figure 24 shows a comparison of the meshes obtained in each software for the second (M2) level of refinement.

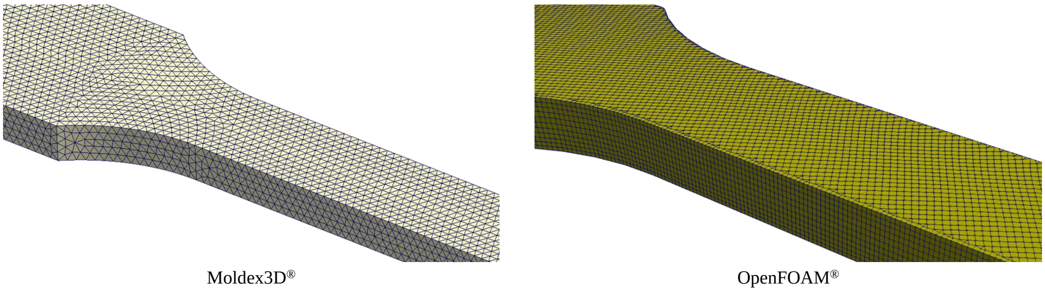

Figure 24: Second level of refinement for Moldex3D® (left) and for OpenFOAM® (right).

Figure 24 shows that although the difference between the total number of cells for this level of refinement is 260 cells, the density of cells in the OpenFOAM® software seems to be clearly higher than in the proprietary software. This is a reason for instead of comparing the same levels of refinement, the idea is to search for converged solutions.

5.4.3. Materials properties

As in the other two case studies, the GPPS - General Purpose Polystyrene Styron 678, from Americas Styrenics was used, the fluid rheological model applied was the Cross-WLF, and for equation of state, the compressible variation of the Modified Tait model was used. The constants for the Cross-WLF model and the coefficients used in the modified Tait model are the ones presented in Appendix E. Thus, the variation of the viscosity as a function of shear-rate and temperature for this material is shown in Figure 22, and the variation of the specific volume with both temperature and pressure is given by the coloured lines presented in Figure 19.

5.4.4. Initial and boundary conditions

The initial and boundary condtions used in this Case Study are presented in Table 13 for OpenFOAM®. The only difference concerning the other case studies is the injection velocity defined at the inlet face, which is U=1.4 m/s.

Table 13: Initial and boundary conditions for OpenFOAM® software in the Case Study 3.

	Temperature (T)	**Velocity (U)**	**Pressure (p)**	**phase (α)**
Inlet	230 °C	1.4 m / s	Null normal gradient	1
Outlet	Null normal gradient	Null normal gradient	10^5 Pa	Null normal gradient
Walls	Heat convection T_{mould} = 50 °C h = 1250 (W/(m^2 K))	0 m / s	Null normal gradient	Null normal gradient
Initial Condition	230 °C	0 m / s	10^5 Pa	0

Moldex3D® being a commercial software, many features are not available for users to define, therefore, sometimes it may be difficult to assure that the bounday conditons between both softwares are the same. Anyway, the flow rate imposed in Moldex3D is the same of OpenFOAM®, which is $Q = 15.7$ (cm^3/s). And the temperature boundary conditions are the same as well. The switch-over point was defined to happen at 98% of cavity volume filled for both softwares. The pressure field could not be accessed in the proprietary software, and therefore, is the only boundary condition that is unknown.

6. Results and Discussion

6.1. Case Study 1: Filling of a cylindrical cavity with a Newtonian fluid

As previously mentioned, this first Case Study is one of the simplest case studies that can be carried out, and was employed to verify if the solver and the numerical procedure are well defined and implemented. For this, a Newtonian fluid was chosen, for which there is a simple analytical solution that allows calculating velocity profiles. Thus, one of the first tests carried out was to analyse the moment when the switch-over point is reached, and to compare it with the theoretical value given by the Equation (81). Therefore, since the analytical solution assumes material incompressibility, Table 14 shows the moment that switch-over is reached for the *openInjMoldSim*, using incompressible variation, with a comparison with the analytical (A) value.

Table 14: Comparison of the switch-over time obtained with the *openInjMoldSim* using incompressible variation with the analytical (A) value.

Mesh	*Switch-over time [s]*	*Error [%]*
M1	0.0249	1.630
M2	0.0246	0.408
M3	0.0230	6.120
A	0.0245	—

Table 14 shows that for the coarse (M1) and medium (M2) levels of refinement the switch-over time is close to the expected, with the differences justifiable by the numerical approximations, specially for M2 level where the error is around 0.4%. For the most refined level (M3), the switch-over point is reached faster than it was expected, resulting in an error of 6%. Therefore, different tests were performed in order to understand the deviation presented in M3. In the end, and since the problem was in the progression of the flow front, a different variation of *openInjMoldSim* was used, where the artificial interface compression term from Equation (76) was removed, named *noInterfaceCompression* solver (NIC), which was presented in the Chapter 5 (Section 5.1). The same results obtained with the new solver are presented in Table 15 for both the regular variation (IC), and the one neglecting the artificial compression (NIC).

Table 15 shows that the IC gives good results for coarse and medium meshes because it minimizes the effect of the interface diffusivity, and, therefore, favours the mass conservation. Since NIC omits the artificial interface compression, the results deviate from the analytical values, especially for the coarsest mesh. When analysing results obtained with refined meshes, one can conclude that the ommition of the artificial interface

Table 15: Comparison of the switch-over time obtained with the *openInjMolSim*, considering interface compression (IC), and neglecting the interface compression (NIC) with the analytical (A) value.

Mesh	*IC*		*NIC*	
	Switch-over time (s)	*Error (%)*	*Switch-over time (s)*	*Error (%)*
M1	0.0249	1.63	0.0462	88.6
M2	0.0246	0.408	0.0250	2.04
M3	0.0230	6.12	0.0247	0.816
A	0.0245	—	0.0245	—

compression term does not affect the mass balance, and ended up filling the cavity near the analytical value. Figure 25 shows the moment of the flow front progression inside the cavity where the filling stage should be reaching at the switch-over point, using both (IC) and (NIC) solvers.

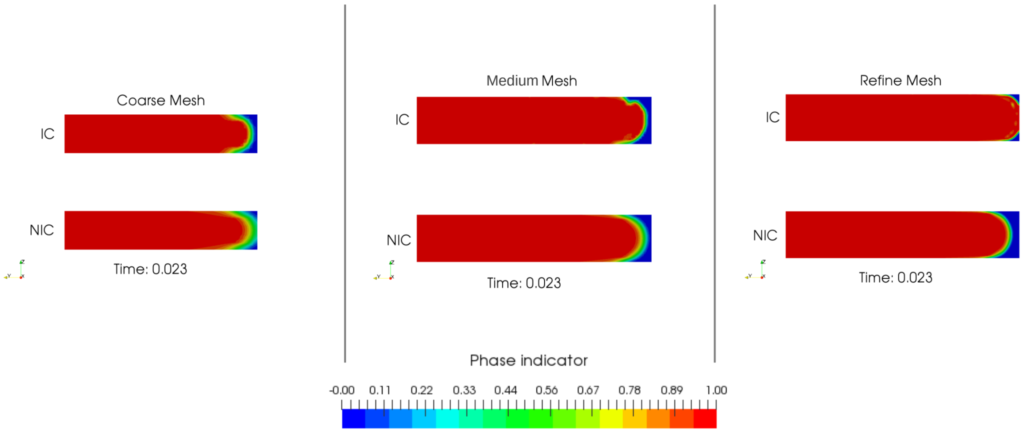

Figure 25: Flow front location at t=0.023s for the different meshes and formulation IC/NIC.

The results shown in Figure 25 allowed to conclude that for coarse meshes (M1) and not considering the artificial interface compression (NIC) the flow front has already reached the outlet, although, as the diffusivity of the interface is larger, the 98% of cavity volume filled in only reached later. The artificial interface compression allows a shaper interface and, therefore, provides values closer to the analytical ones. In medium meshes (M2) due to the mesh refinement, the flow front becomes sharper for both variations, and the difference between them are low as shown in Table 15. For the refined meshes (M3), Figure 25 shows that the artificial interface compression term makes the solver filling the cavity sooner than it was expected, as shown in Table 15, however, for the NIC solver the cavity is filled 0.0015 seconds later than the time represented in Figure 25 respecting the analytical solution. The reason for this difference in refined meshes may be from the fact that the natural refinement of the mesh may be sufficient to promote a sharper interface and, therefore, the artificial interface compression term is not required.

To further investigate the effects of the artificial interface compression in the *openInjMoldSim* solver, an analytical solution for the calculation of the velocity profiles was employed, and, they were analysed for both variations, *i.e*, considering the interface compression and not considering it. The analytical solution was presented by Liang [115] for the power-law model and considering $n = 1$ we obtain the equivalent equation for Newtonian fluids given by

$$U(r) = \frac{r_0}{2}\left(\frac{r_0 \Delta p}{2\mu L}\right)\left[1 - \left(\frac{r}{r_0}\right)^2\right] \tag{87}$$

where $\frac{\Delta p}{L}$ is taken as the pressure drop throughout the channel length.

Figure 26 shows the velocity profiles throughout the radius of the cylindrical cavity for all levels of refinement using both *openInjMoldSim* (IC) and the (NIC) solvers. Figure 26 presents the x-axis normalized by the radius of the channel (r_0 = 0.010 m), and the y-axis normalized by the inlet velocity (U_0 = 4 m/s).

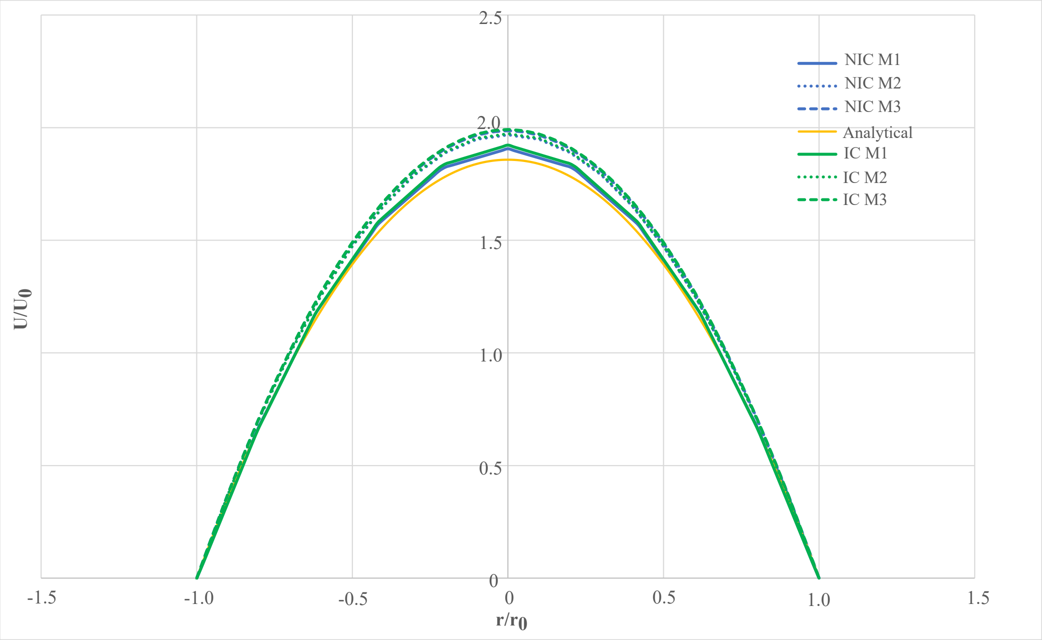

Figure 26: Velocity profiles taken through the radius of the cylindrical cavity for the three levels of refinement of both *openInjMoldSim* (IC) and *noInterfaceCompression* (NIC) solvers.

Figure 26 shows that the maximum velocity happens at the centre of the channel, and at the walls it is null due to the no-slip boundary condition applied. Figure 26, also shows that the velocity profiles obtained with both variations are very similar between each other, especially for the second and third levels of refinement. Comparing with the analytical solution, the profiles are not exactly the same, because the analytical solution does not consider temperature effects, and, it was considered in the numerical calculation.

To end this study of the effects that the artificial interface compression has in the *openInjMoldSim*, the Richardson's extrapolation method (Section 5.1.2) was applied to velocity values obtained in the previous result, and to the pressure values obtained in that location as well, to estimate the errors for each level of mesh refinement, and to compare the results between both IC and NIC. Table 16 presents the results for the velocity values, and Table 17 presents the results for the pressure field.

Table 16: Velocity values for the mesh refinement study for both IC and NIC solvers, Richardson's extrapolation (RE), and the relative errors.

Mesh	*IC*		*NIC*	
	U/U_0	*Error (%)*	U/U_0	*Error (%)*
M1	1.922	4.00	1.906	4.43
M2	1.973	1.48	1.968	1.35
M3	1.992	0.55	1.986	0.41
RE	2.003	—	1.995	—

Table 16 shows that the artificial interface compression term does not influence the results obtained for velocity field, since for the same level of refinement, the values are very close.

Table 17: Pressure values for the mesh refinement study for both *openInjMoldSim* and *noInterfaceCompression* solvers, Richardson's extrapolation (RE), and the relative errors.

Mesh	*IC*		*NIC*	
	Pressure (MPa)	*Error (%)*	*Pressure (MPa)*	*Error (%)*
M1	4.61	0.11	4.46	5.93
M2	4.65	0.98	4.61	2.77
M3	5.01	8.79	4.68	1.29
RE	4.61	—	4.74	—

The pressure values shown in Table 17 presented a different behaviour from the ones for the velocity. For IC the mesh refinement did not produce an asymptotic behaviour for the pressure field, and, therefore, the error increased with mesh refinement. However, using the NIC variation, the pressure values presented the asymptotic behaviour, and the errors in each level of refinement decreased towards the extrapolated one. All of these results give the idea that using the VOF method without the artificial interface compression, and letting the natural refinement mesh study comprise the interface might be a good solution for multiphase flows.

Bearing in mind the two variations of the modified Tait model for the equation of state presented in Chapter 5, the same set of results were analysed using both, but this time using only the using NIC solver, because it was the solver that gave the best results in the previous study. The objective is to understand the differences caused by considering the polymer melt as incompressible, since the compressible model is the most realistic one. Table 18 shows the switch-over time obtained for both variations for different mesh refinement levels.

Table 18: Comparison of the switch-over time obtained with the compressible (reference), and incompressible variations of the Modified Tait model using NIC solver.

Mesh	Switch-over time (s)		*Relative*
	Compressible (Reference)	*Incompressible*	*difference (%)*
M1	0.0462	0.0462	0.0
M2	0.0249	0.0250	0.402
M3	0.0246	0.0247	0.407

Table 18 shows that the results for both variations, compressible and incompressible, are close to each other. In fact, only for the M2 and M3 levels that the switch-over time was different, with the compressible variation filling the cavity slightly faster than the incompressible one. This may be explained from the fact that as pressure is high at the inlet face, the material tends to compress at that location. Being the volumetric flow rate imposed at the same place, therefore, more mass is injected inside cavity for the compressible variation. As the material flows, and advances in the cavity, the pressure decreases, and, therefore, the material tends to expand, which results in a higher volume fraction occupied, and, consequently, in a faster filling of the cavity, although the differences are really small.

Applying Equation (87) for both variations, the velocity profiles for both compressible and incompressible variations of the equation of state throughout the radius of the channel was investigated. Figure 27 shows the velocity profiles for all meshes employed in both variations, as well as the analytical solution. Remember that the analytical solution cannot be precisely compared due to the lack of the temperature effect in the calculation.

The results presented in Figure 27 show once again the typical quasi-parabolic velocity profile, being the profiles obtained with each variation close to the analytical one, although they cannot be precisely compared. Moreover, the differences for the velocity profiles obtained with both variations of the Modified Tait model are very small, and, therefore, cannot be depicted. The Richardson's extrapolation (Section 5.1.2) was once again employed in the three levels of refinement for both the velocity values obtained in the center of the channel of Figure 27, and for the pressure field in that location. The differences between each variation were compared, taking the compressible as the reference. Table 19 shows the velocity values for all meshes, the extrapolated value

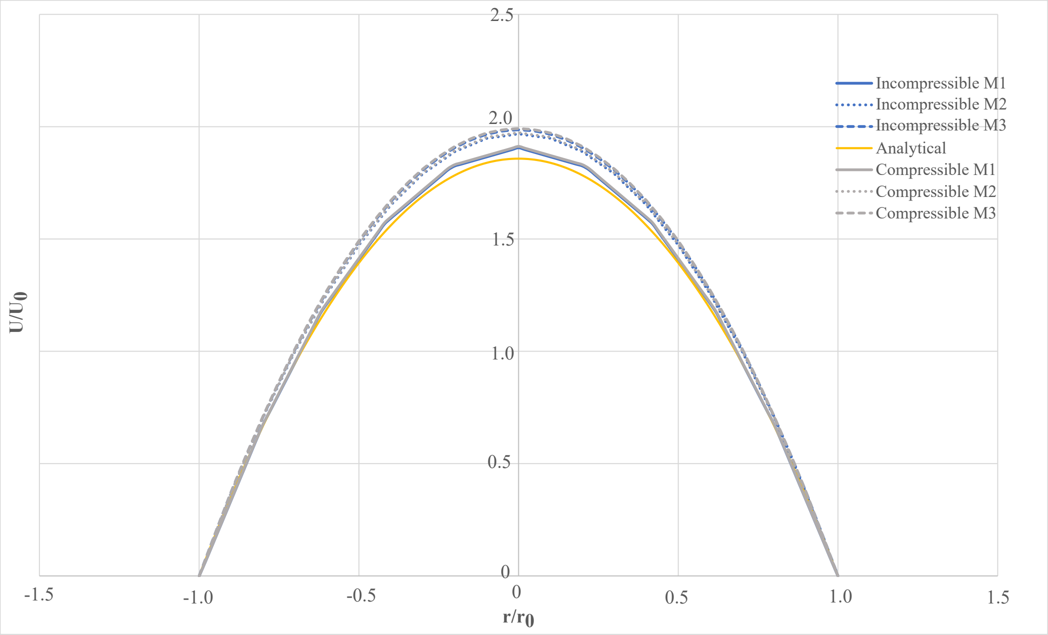

Figure 27: Velocity profiles throughout the radius of the cylindrical cavity for both Compressible and Incompressible variations for the mesh refinement study, and a comparison with the analytical (A) one.

obtained using Richardson's extrapolation (Equation (84)) and the errors (Equation (85)) for each level of refinement using both compressible and incompressible variations.

Table 19: Velocity values for the mesh refinement study for both compressible and incompressible variations, Richardson's extrapolation (RE), and the relative errors.

Mesh	*Compressible*		*Incompressible*	
	U/U_0	*Error (%)*	U/U_0	*Error (%)*
M1	1.912	4.37	1.906	4.43
M2	1.973	1.33	1.968	1.35
M3	1.992	0.40	1.986	0.41
RE	2.000	—	1.995	—

Table 19 shows that the velocity values present a monotonic and asymptotic behaviour with mesh refinement for both variations, and, therefore, the errors for each level of refinement decreases. Moreover, the velocity values are very close between both models, and that is why no differences could be depicted in Figure 27. Table 20 shows the same procedure but for the pressure values.

Table 20 shows that just like velocity values, the pressure values present a monotonic and asymptotic behaviour, with the errors with each level of refinement decreasing towards the extrapolated value. Although, the values of pressure are very close between both variations, the compressible one presented always higher values, which agrees with the fact

Table 20: Pressure values for the mesh refinement study for both compressible and incompressible variations, Richardson's extrapolation (RE), and the relative errors.

Mesh	*Compressible*		*Incompressible*	
	Pressure (MPa)	*Error (%)*	*Pressure (MPa)*	*Error (%)*
M1	4.49	5.96	4.46	5.93
M2	4.65	2.61	4.61	2.77
M3	4.72	1.14	4.68	1.29
RE	4.77	—	4.74	—

of in this variation, the cavity being filled quicker than in the incompressible one.

From the results presented, we conclude that incompressible variations do not affect significantly the accuracy of the final results. Later, this analysis will be extended to the Cross-WLF rheological model, which is usually used in the filling stage of the injection moulding process.

6.2. Case Study 2: Effects of compressibility on the filling of a rectangular cavity with a *Cross-WLF* fluid.

This case study aimed to assess the fluid compressibility effects in the filling stage of the injection moulding process. The results obtained using all the meshes considered (see Table 9) do not present substantial differences, therefore, only the ones obtained with refined meshes (M3) are presented.

Table 21 shows the execution time of the injection moulding solver to simulate the filling phase of a rectangular cavity using both compressible (C) and incompressible (I) variations of the Modified Tait model for the equation of state.

Table 21: Execution time spent by each variations of the Modified Tait model and the relative differences between them for the filling stage of a rectangular cavity.

Variation	*Mesh*	*Switch-over time (s) (t_{SO})*	*Execution Time (s)*	*Relative time (%)*
Compressible (reference)	M3	0.98	150 951	—
Incompressible		0.98	143 623	-4.9
Analytical		0.98		

As expected the incompressible variation is 5% faster than the compressible variation, due to the lower interdependence between the flow variables (T, p, ρ). Regarding the evolution of the physical variables during the filling phase, Figure 28 shows the inlet pressure profile over time, for both compressible and incompressible variations. The time axis is normalized by the switch-over time ($t_{SO} = 0.98$ s) presented in Table 21.

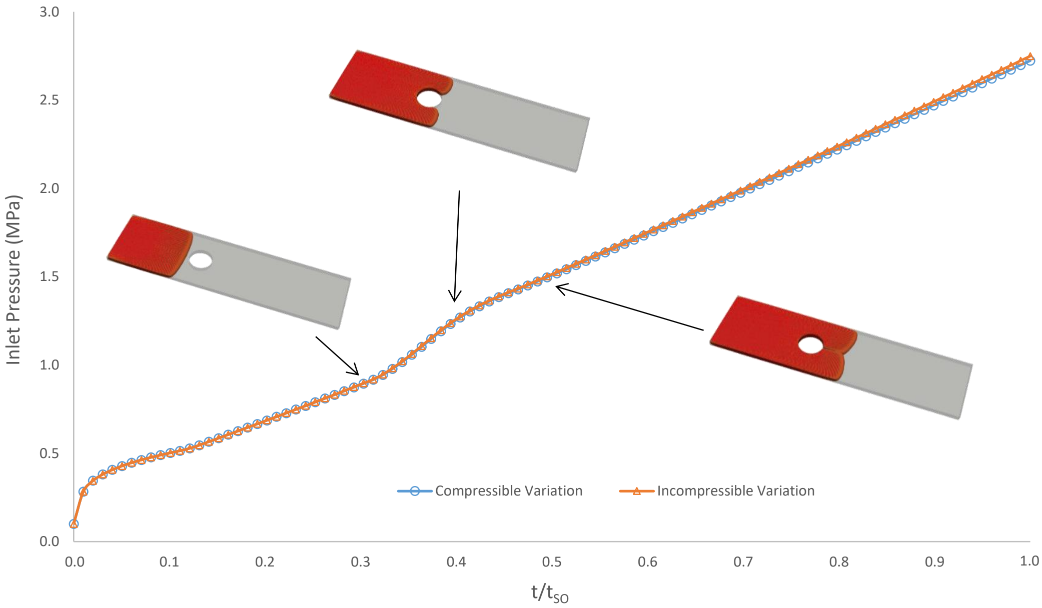

Figure 28: Evolution of inlet pressure over time for the two variations of the Modified Tait for the equation of state in rectangular cavity.

Figure 28 shows that both variations of the Modified Tait model present the same pressure evolution. The pressure evolution presents a constant slope until the flow is divided by the cylindrical insert, because the cross-section is constant until then. During the passage through the cylindrical insert, the pressure slope increases due to the smaller cross-section for the material to flow. After the cylindrical, the initial slope for the pressure evolution is resumed.

Velocity and temperature contours were taken from two cross-sections (slices) at different geometry locations, see Figure 29, representative of the initial flow conditions (S1: x = 40 mm), and melt flow front at cylindrical insert (S2: x = 55mm). These results were investigated for the switch-over point, which represents the end of the filling phase.

Figure 30 shows the velocity and temperature contours for the slices presented in Figure 29, for both variations of the Modified Tait model.

Figure 30 shows that in the first slice, as this one is close to the cylindrical insert, the melted material at the centre of the channel flows at a lower velocity due to the cylindrical insert than the material that flows between the centre of the channel and the mould walls. This means that the first slice (S1) shows the beginning of the division of the flow front. The minimum velocity is presented at the mould walls due to the no slip boundary condition employed. Comparing both variations, there are no visible differences between each other.

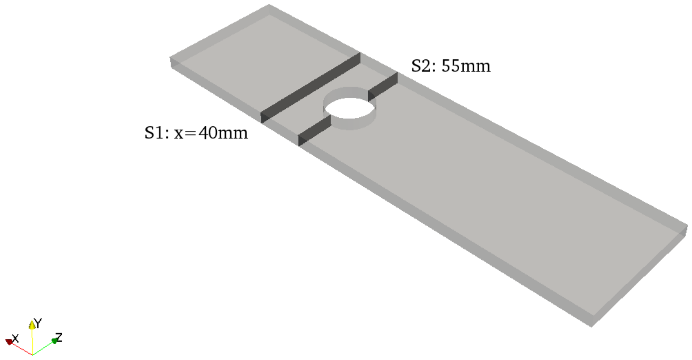

Figure 29: Locations for the analysis of the velocity and temperature contours.

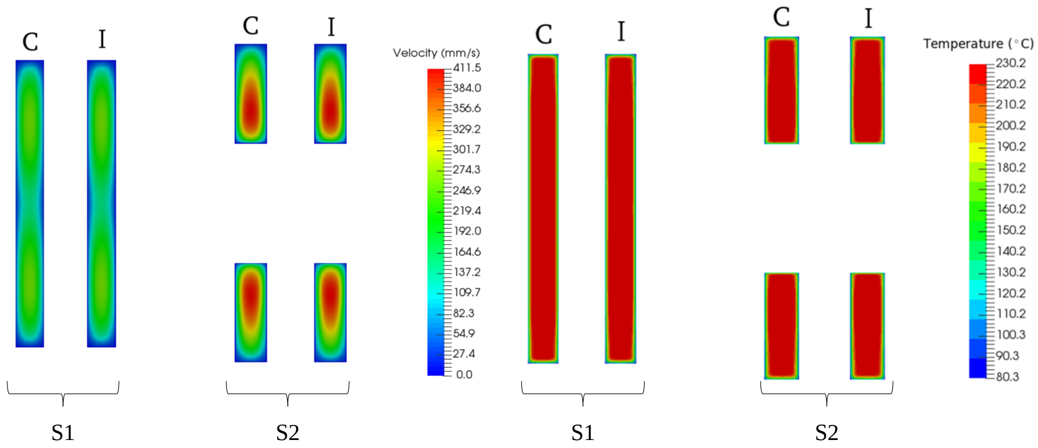

Figure 30: Velocity (left) and temperature (right) contours for both slices in the rectangular cavity, C - compressible variation, I - incompressible variation.

In terms of temperature, as it was expected, the centre of the channel is hotter than the walls. However, the minimum temperature is higher than 50°C considered at the beginning due to the heat transfer from the polymer melt to the mould walls. Comparing both models, there are no visible differences between them, which means that a more detailed test is needed. The second slice shows an axisymmetric velocity profile because the reduction in cross section is the same on both sides of the cylindrical insert. As you would expect, the velocity increases as the material has a smaller section to flow through. However, the two variations of the Modified Tait model showed no visible differences between them. The temperature field did not presented too much differences between each slices, temperature in the middle of the channel is still higher than the moulds walls. Although the cross-section decreased, the maximum temperature was around 230°C, defined at the beginning, meaning that the viscous dissipation in very small in this case. The variations of the Modified Tait model employed did not presented visible differences once again.

In order to measure the differences imposed by the incompressible variation of the equation of state, the minimum, average, and maximum values for both fields (temperature and velocity) were computed at both locations on the switch-over point, being the results presented in Table 22 and 23.

Table 22: Minimum, average and maximum temperature values at slices S1 and S2 at the switch-over point.

Location	*Variations*	*Temperature (°C)*			*Relative differences (%)*		
		T_{min}	T_{ave}	T_{max}	T_{min}	T_{ave}	T_{max}
S1	C (ref)	109.3	218.4	230.1	—	—	—
	I	108.6	218.3	230.2	0.6	0.1	0.1
S2	C (ref)	112.9	216.2	230.0	—	—	—
	I	112.3	216.1	230.2	0.5	0.1	0.1

Table 23: Average and maximum velocity values at slices S1 and S2 at the switch-over point.

Location	*Variations*	*Velocity (mm/s)*		*Relative differences (%)*	
		U_{ave}	U_{max}	U_{ave}	U_{max}
S1	C (ref)	148	257	—	—
	I	149	258	0.7	0.4
S2	C (ref)	233	409	—	—
	I	233	411	0.0	0.5

Notice that the minimum value of the velocity field is not presented in Table 23 because on the mould walls the velocity boundary condition applied was no-slip, which means that for every case, the minimum velocity is zero. From the results shown in Tables 22 and 23 we conclude that the differences between the compressible and incompressible variations in both velocity and temperature fields are small, *i.e*, smaller than 1%.

Regarding the behaviour of the flow fields along all the geometry length, Figure 31 shows the distribution of the pressure field computed for both compressible and incompressible variations.

As Figure 31 shows the maximum pressure needed to fill the cavity for both variations is practically the same, with the difference being only of 3.7%, which means that the differences are minimum. Notice that the minimum pressure is the same for both variations because in the outlet boundary condition, the pressure was defined as atmospheric pressure, therefore, the minimum value presented in Figure 31 is the same for both variations of the Modified Tait model. Therefore, this result confirms once again the similarities in the results obtained through a compressible and incompressible variations.

From all the results presented, we can conclude that both incompressible and

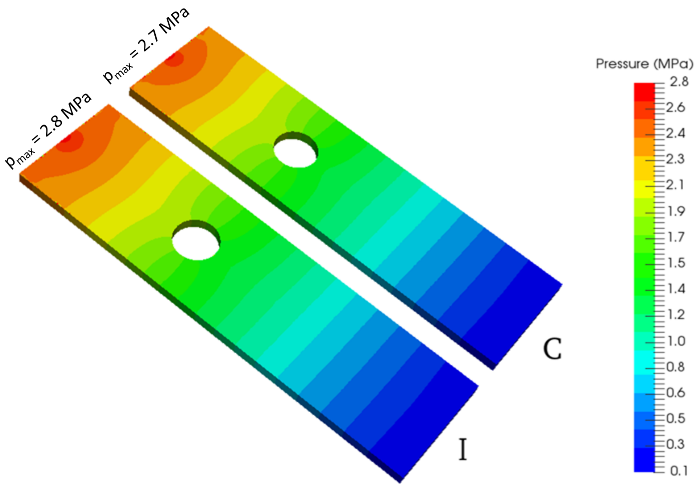

Figure 31: Evolution of the pressure throughout the part in the two variations, C - compressible variation, I - incompressible variation.

compressible variations are adequate to be used on the simulation of the filling stage of the injection moulding process, with the former being more advantageous in terms of computational cost. Logically, the creation of truly incompressible formulations and the comparison with the compressible one would have a great importance in order to end the discussion about the material model to employ in the filling stage of the injection moulding process.

6.3. Case Study 3: Filling of a tensile test specimen.

As stated on Section 5.4 the results here presented for the filling of a tensile test specimen are computed with the open-source code *openInjMoldSim* and the commercial software *moldex3D*®. At a first step we performed a mesh sensitivity analysis using both softwares. This way, we expect to obtain accurate solutions which do not depend on the mesh size. To verify the level of refinement needed to ensure the accuracy of the solutions obtained, the Richardson's extrapolation [139], described in Section 5.1.2, is applied for all the field variables studied (p, T, and U).

Table 24 shows the values of the maximum pressure concerning all cavity, the respective extrapolated value obtained through Richardson's extrapolation (RE) (Section 5.1.2), and the associated errors for both softwares. The values were measured for the switch-over point, which means the end of the filling phase.

The extrapolated value for Moldex3D® was made using Equation (86) because the calculated apparent order, through Equation (83), gave illogical results due to the fact that the errors did not converge assymptotically with mesh refinement. Therefore, the apparent order for the proprieatry software was estimated as 1.5 due to the order of the schemes used in the calculation [8]. An example of the results obtained with apparent order calculated in

Table 24: Maximum pressure values obtained with the commercial software Moldex3D® and the open-source software OpenFOAM® for the switch-over point. The Richardson's extrapolated (RE) values and the relative errors are also presented.

Mesh	Moldex3D®		OpenFOAM®	
	Maximum (MPa)	*Error (%)*	*Maximum (MPa)*	*Error (%)*
M1	7.18	43.4	7.76	29.6
M2	8.32	34.4	14.07	27.8
M3	9.66	23.8	12.01	9.1
M4	11.61	8.41	—	—
RE	12.68	—	11.01	—

Moldex3D® is given in Appendix I. The OpenFOAM results were extrapolated using the usual Richardson's extrapolation given by Equation (84), with the apparent order calculated by Equation (83).

From Table 24 we understand the reason for the existence of four levels of refinement in Moldex3D®. For this software, the error presented at the third level of refinement is still very high (>20%), therefore, a fourth mesh was employed, and, the error for this new level of refinement is above 8%. This result means that the proprietary software need at least four degrees of refinement to obtain an accurate enough result for the pressure field [132]. The pressure results obtained with the OpenFOAM® software (see Table 24) are more accurate than the ones obtained with Moldex3D® for the same level of refinement. And, it needed one less level of refinement to have the same magnitude of errors of the Moldex3D® software. Notice that the computed maximum values are more difficult to control, thus, tend to have larger errors, due to the fact that they are local values, and therefore, they have higher sensibility to perturbations.

Figure 32 shows the contour of the of the pressure evolution in the tensile test specimen part. As shown in Figure 32, the pressure profiles obtained on both softwares are qualitatively identical, when using the most refined meshes for both of them (M4 on Moldex3D® and M3 on OpenFOAM®). However, the flow front in the open-source software seems more realistic than the commercial one where the flow front seems to progress like a flat.

To complete the mesh refinement study, and the effect that the mesh cell size has on the final results, the same procedure employed for the pressure field was extended to the temperature and velocity fields. Table 25 shows the values of maximum temperature for each level of refinement using both softwares, the extrapolated value (RE) given by Richardson's extrapolation (Section 5.1.2), and the relative errors. The values were measured for the switch-over point.

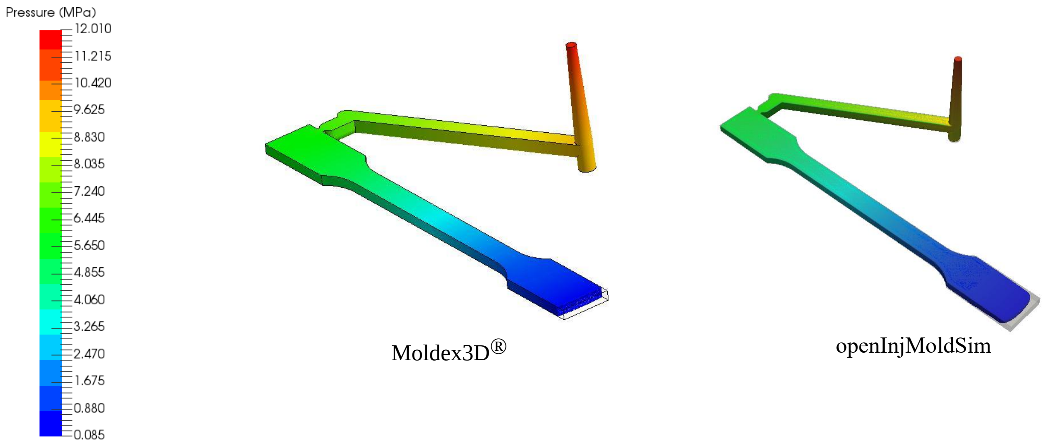

Figure 32: Evolution of pressure from inlet until switch-over point for the most refined meshes of both softwares.

Table 25: Maximum temperature values obtained with the commercial software Moldex3D® and the open-source software OpenFOAM® for the switch-over point. The Richardson's extrapolated (RE) values and the relative errors are also presented.

Mesh	Moldex3D®		OpenFOAM®	
	Maximum (°C)	*Error (%)*	*Maximum (°C)*	*Error (%)*
M1	230.2	0.7	230.0	1.0
M2	230.3	0.6	232.2	0.1
M3	230.5	0.5	232.4	0.0
M4	231.3	0.2	—	—
RE	231.7	—	232.4	—

The problem presented for the pressure field obtained with Moldex3D® was also presented in the temperature. Therefore, instead of extrapolating the value of the maximum temperature with three meshes, and a calculated apparent order, the same procedure employed before for the pressure field was used in here as well, meaning that the apparent order was assumed as 1.5, and the extrapolated result given by Equation (86).

Table 25 shows that in general manner, both softwares can deal well with temperature, and, even for the coarsest levels, the errors are already below 1% for both of them. We conclude that the viscous dissipation is increasing the melt temperature in 2°C, which do not degrade the material since the range of processing temperatures is $180\ °C < T_{melt} < 280\ °C$ [66]. Figure 33 shows the temperature contours for the slice (S1) presented in Figure 23.

Figure 33 shows non-physical contours on the core region for the proprietary software, which can be due to the existence of larger cells on that location, however, Moldex3D® does not allow a clear analysis of the mesh in that section. The OpenFOAM® contours are

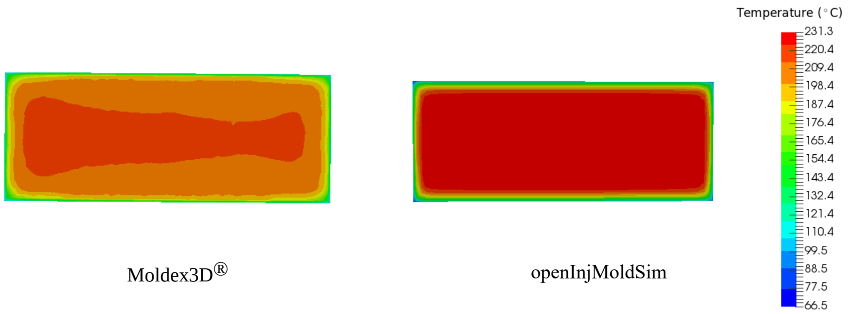

Figure 33: Temperature contour for the most refined meshes of both softwares.

much more realistic and smoothed. Table 26 presents the maximum velocity values for each level of refinement using both Moldex3D® and OpenFOAM®, the extrapolated value (RE) given by Richardson's extrapolation (Section 5.1.2), and the relative errors. The values were measured for the switch-over point.

Table 26: Maximum velocity values obtained with the commercial software Moldex3D® and the open-source software OpenFOAM® for the switch-over point. The Richardson's extrapolated (RE) values and the relative errors are also presented.

Mesh	Moldex3D®		OpenFOAM®	
	Maximum (mm/s)	*Error (%)*	*Maximum (mm/s)*	*Error (%)*
M1	2080	14.2	2530	33.2
M2	1950	19.6	4800	26.7
M3	1930	20.4	4100	8.2
M4	2250	7.22	—	—
RE	2430	—	3790	—

As happened with pressure and temperature fields, due to the none convergence of errors in Moldex3D®, instead of extrapolating the value through Equation (84), and calculating a apparent order through Equation (83), only the two more refined meshes are used to extrapolate the velocity values using Equation (86) assuming a apparent order of 1.5.

Just like what happened with pressure field, the proprietary software needed one more degree of refinement to obtain the same order of magnitude of errors that OpenFOAM® presents. And even for this fourth mesh of refinement, the erros in Moldex3D® are superior to 7%. The error obtained for the maximum velocity in M3 of OpenFOAM® are above 8%. However, it is important to notice that maximum values are more difficult to control. Anyway, from Tables [24-26] we can conclude that the velocity field is the one presenting more inconsistent results between both softwares, there is a difference of 1000 (mm/s) between the Richardson's extrapolated values. Figure 34 shows the velocity contours for the slice (S1) presented in Figure 23.

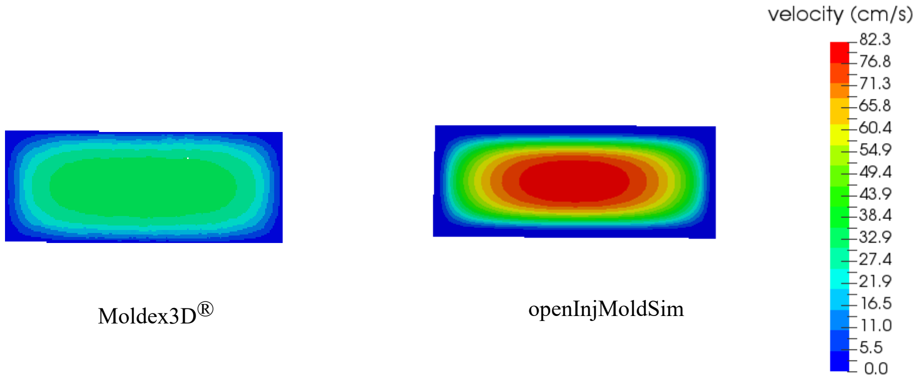

Figure 34: Velocity contour for the most refined meshes of both OpenFOAM® and Moldex3D®.

Figure 34 shows the inconsistency between the results obtained with both softwares for the velocity field that were also presented in Table 26. The velocity of 38 (cm/s) in the center of the channel for Moldex3D® is very different of the 82 (cm/s) of OpenFOAM®. Although, knowing that the volumetric flux is 15.7 (cm^3/s), the average velocity in that section should be around 40 (cm/s). This means that the result obtained with Moldex3D® is completely wrong because the maximum velocity taken from Figure 34 is around 40 (cm/s), which should be the average velocity, and, since the walls are not animated of velocity, the average value for that cross-section is below the theoretical one. For a Newtonian fluid, and in a parallel plates flow, the maximum velocity is 1.5 times higher than the average on the section [140]. The contours of OpenFOAM® seems to be more realistic due to the fact that the maximum velocity in the center of the channel is around the double of the average value (80 (cm/s)). If we look at the first case study, although we are working with Newtonian fluids, and we do not estimate the effect of temperature and shear-rate on viscosity, the velocity in the center of the channel was around 1.8 times higher than average one, which give may give reason to the fact that the velocity profile of the open-source software is the corrected one.

To fix the inconsistency of the velocity values presented by Moldex3D® we envisage to simulate with Moldex3D® a channel that had the same length of the tensile test specimen, and also the same width and thickness of the most restricted channel region (see Figure 35). The processing conditions and material used were the same as before, and four levels of refinement were employed. A slice (S1) at the same location was used to obtain the velocity field contours at the switch-over point for all levels of refinement (see Figure 35).

Figure 36 shows the velocity contour taken at the switch-over point for all meshes employed in the central channel of the tensile test specimen case study with Moldex3D®.

Figure 36 shows that with the mesh refinement, the maximum velocity in that cross-section increases, with the maximum velocity being around 30 cm/s for M1, 40 cm/s

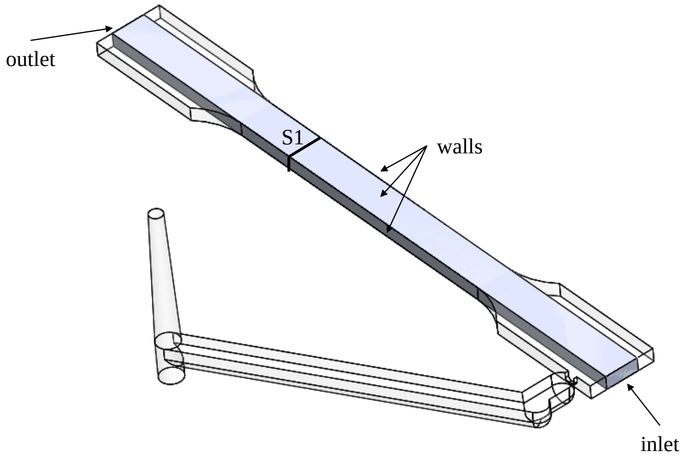

Figure 35: Central channel of the tensile test specimen.

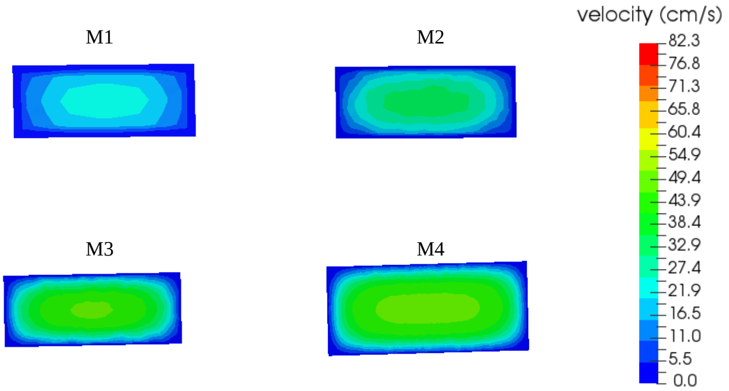

Figure 36: Velocity contour for the mesh refinement study employed in the central channel of the tensiles test specimen obtained with Moldex3D®. The contours were taken at the switch-over point.

for M2, 55 cm/s for M3, and 60 cm/s for M4. This means that with the mesh refinement, the contours become more approximate from the ones represented in OpenFOAM (See Figure 34). With the results presented for the three field variables, we conclude that the experimental assessment of these results is needed to understand which software gives the better prediction, but it seems that the OpenFOAM® has the better accuracy, at least theoretically.

Finally, the inlet pressure evolution along the time was monitored and compared for the OpenFOAM® and Moldex3D® softwares for each level of refinement (see Figure 37). These results will allow, in the future, a direct comparison with the experimental data and, therefore, further validation of both softwares.

Figure 37 presents some perturbations periods in the evolution of the pressure. These perturbations will be compared with the progression of the flow front in each period. The intervals where pressure presented oscillations happened for $0.1 < t/t_{SO} < 0.2$, $0.3 < t/t_{SO} <$

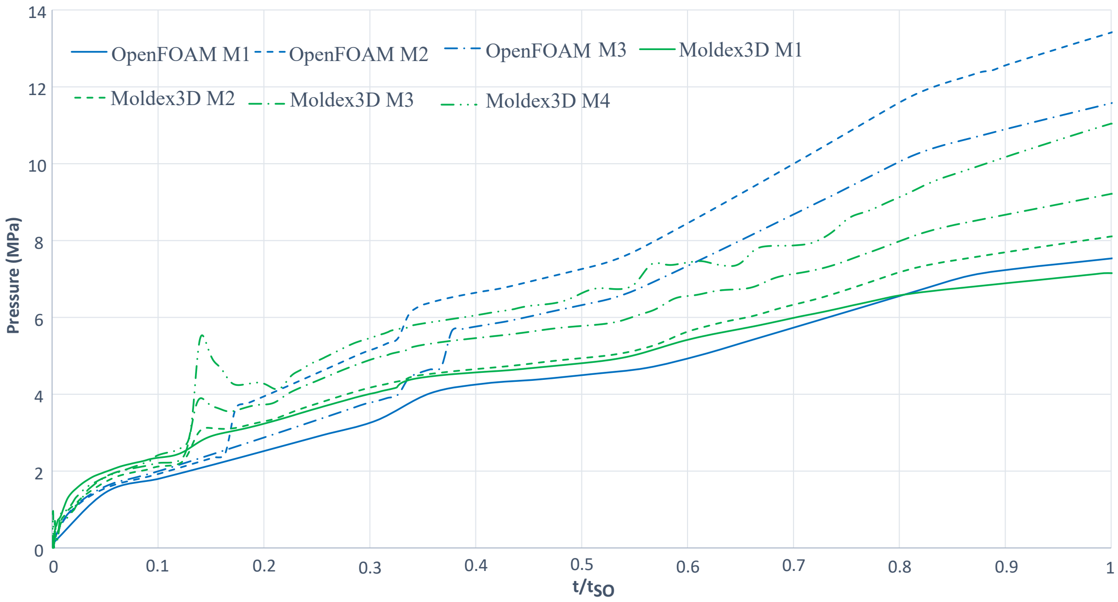

Figure 37: Time history of the inlet pressure at different mesh size resolutions obtained with both OpenFOAM® and Moldex3D® softwares.

0.4, and $0.5 < t/t_{SO} < 0.8$, as shown in Figure 37. The first interval is represented by the melt front time shown in Figure 38. This interval corresponds to the passage of the melt from the sprue to the main runner. Notice that the pressure perturbation presented in Figure 37 happens especially for the proprietary software.

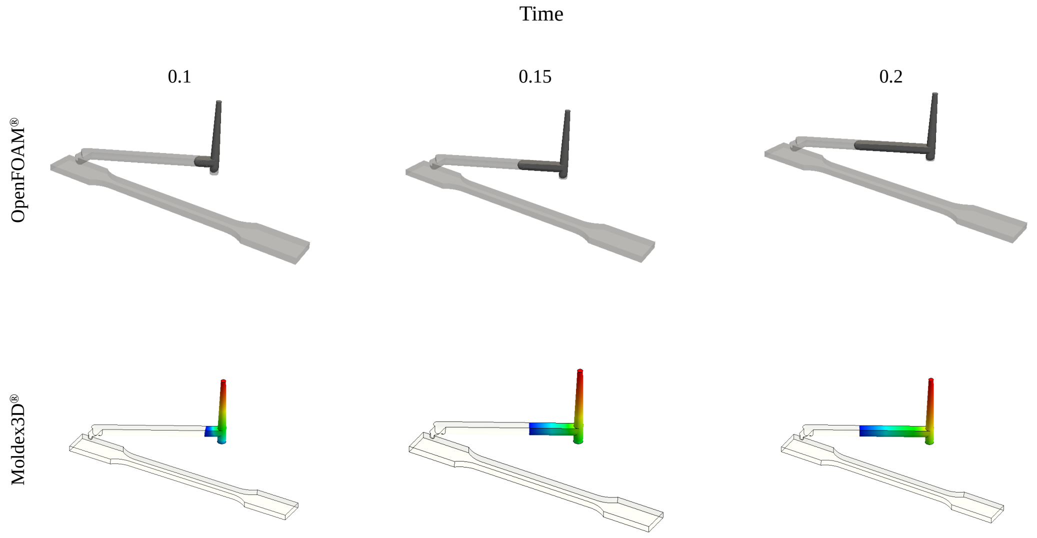

Figure 38: Melt front time between $t/t_{SO} = 0.1$ and $t/t_{SO} = 0.2$ in both OpenFOAM® and Moldex3D®.

Analysing Figure 38 the perturbation presented in Figure 37 can be explained by the fact that the melt is forced to progress in two different directions (the main runner, and the cold slug), and, therefore, an increase of presure could happen. However, the perturbation happened especially for Moldex3D®, and, for the proprietary software, the cold slug is already filled at $t/t_{SO} = 0.1$, which contradicts the justification. The second perturbation found on the inlet pressure evolution (see Figure 37) occurs in the interval represented by the melt front time shown in Figure 39, which coincides with the passage of the melt material through the gate. This second oscillation happens especially for OpenFOAM®, while for Moldex3D® there is a slight increase.

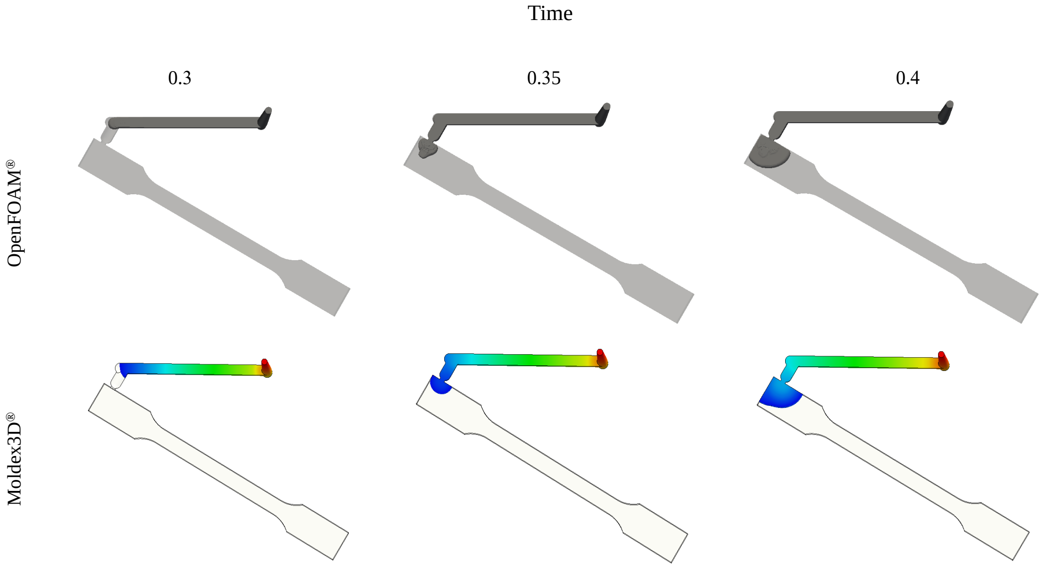

Figure 39: Melt front time between $t/t_{SO} = 0.3$ and $t/t_{SO} = 0.4$ in both OpenFOAM® and Moldex3D®.

As explained in Chapter 2, the gate is a very small and thin channel, meaning that it present a high resistance to the flow, and, consequently, the melted material does not flow so easily [11]. Therefore, more pressure is usually applied in order to make the melt fill this gap, meaning that an increase of the pressure is expected at this region. As stated before, in Figure 37 the different meshes of OpenFOAM® registered an increase of pressure in this interval. However, in Moldex3D®, the coarsest level of refinement presented a slightly increase in pressure evolution, but with mesh refinement, that increase disappear.

The last oscillation presented in the time evolution of inlet pressure (see Figure 37) is represented by the melt front time presented in Figure 40. This interval corresponds to the passage of the melt front by the thinner channel of the tensile test specimen. As the cross-section becomes narrowed in that section an increase of pressure is expected.

Figure 37 shows that for all OpenFOAM® levels of refinement, the slope of the curve is higher in the interval, than it is after. This is due to the fact that between $0.5 < t/t_{SO} < 0.8$ the cross-section becomes narrowed, and after that, the cross-section increases again. In Moldex3D® for the first two levels of refinement, a similar behaviour happens, but in M3 and M4, the pressure evolution curve fluctuates a lot in that interval without a specific reason for that occurrence, however the pressure is still increasing, as expected. With these results, in the future we can compare the values obtained experimentally with the values obtained in this study, and, therefore, find out which software presents a better accuracy, although,

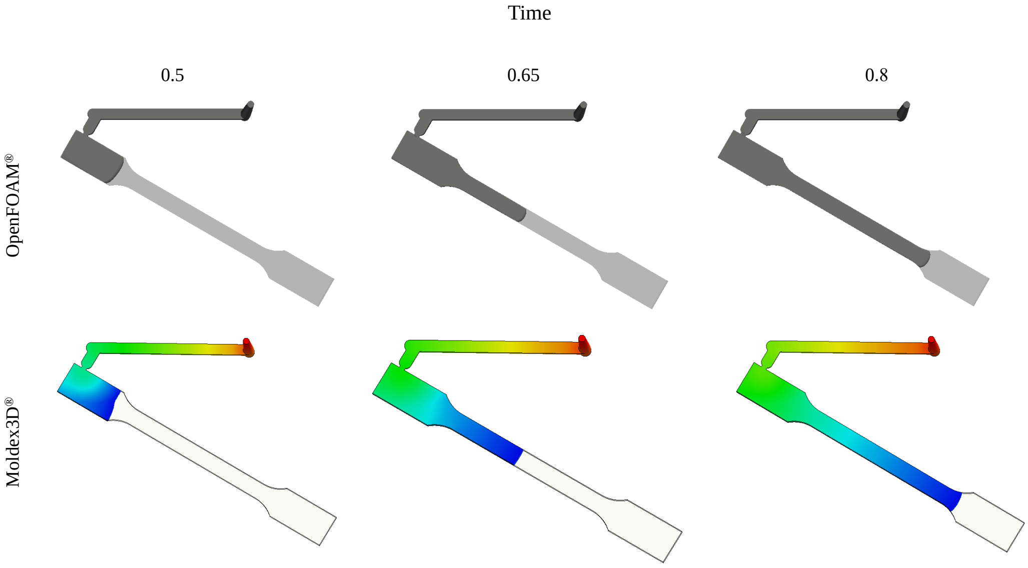

Figure 40: Melt front time between $t/t_{SO} = 0.5$ and $t/t_{SO} = 0.8$ in both OpenFOAM® and Moldex3D®.

theoretically, OpenFOAM® seems to be better.

Note that in Moldex3D® there is no tool that allows to simply see the filling of the cavity without depending on any variable. Therefore, in Figures [38 - 40] the flow front of the Moldex3D® is coloured over time, however, we were unable to replicate this Moldex3D tool in OpenFOAM, and therefore the color scale was not placed in the Figures [38 - 40]. Figures [38 - 40] show that the shape of the flow front between the softwares is different, therefore, Figure 41 presents an approximate view of the flow front in three different regions of the part.

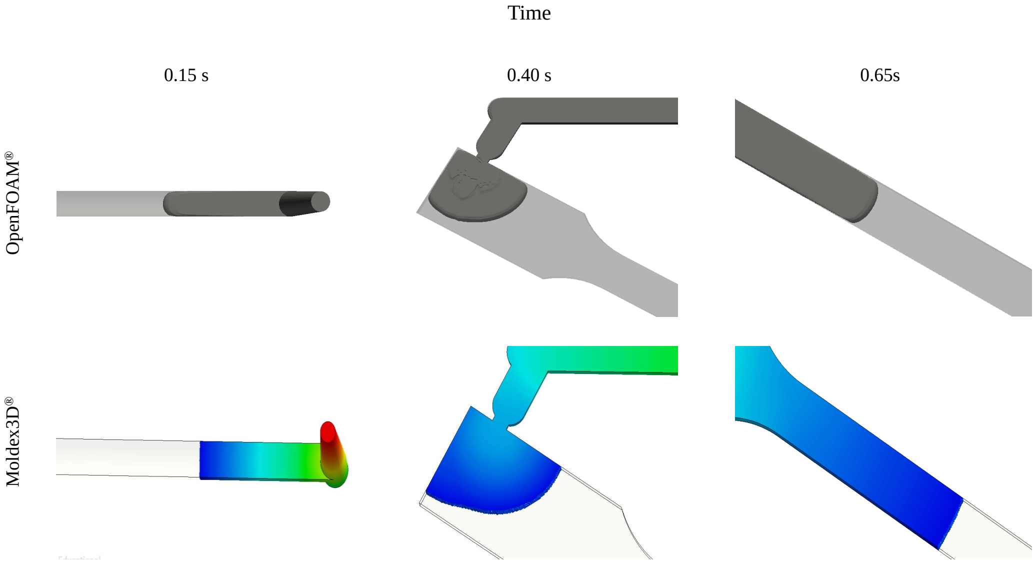

Figure 41: Melt front shape at three different location in both OpenFOAM® and Moldex3D®.

Since the velocity near the mould walls is zero, the shape of the flow front should have be a parabolic type form, meaning that the center flows faster, but the material near the walls is not animated of velocity. Figure 41 shows that the flow front is progressing like a flat in Moldex3D®, with all flow front flowing at the same velocity for the first and third locations, and, in the second one, the flow front presents the expected parabolic format. In OpenFOAM® the flow front has the parabolic format in all locations. This result might come from the use of low order schemes in the proprieatry software, while OpenFOAM® uses second order schemes for the flow front progression. Finally, a general review of the accuracy and performance of both softwares is presented in Table 27, where the errors obtained for the three variables as well as the execution time for each level of refinement are presented.

Table 27: General comparison of accuracy and performance of both softwares, OpenFOAM® and Moldex3D®.

Mesh	*Software*	*Errors (%)*			*Number of*	*Execution*
		p_{max}	U_{max}	T_{max}	*Processors*	*Time*
M1	*Moldex3D®*	43.4	14.2	0.7	1	88 seconds
M2		34.4	19.6	0.6	1	23 minutes
M3		23.8	20.4	0.5	8	3.8 hours
M4		8.41	7.22	0.2	8	24.5 hours
M1	*OpenFOAM®*	29.6	33.2	1.0	1	12.8 hours
M2		27.8	26.7	0.1	8	41 hours
M3		9.1	8.2	0.0	48	98.5 hours
					96	59 hours
					192	34 hours

Table 27 shows that for all levels of refinement OpenFOAM® presents better accuracy than Moldex3D®, with the exception of the temperature field, where both softwares present the same order of magnitude of errors. In order to have the same magnitude of errors for all fields, Moldex3D® needs at least one more degree of refinement than OpenFOAM®, however, it should be remembered that for the commercial software the results did not converge asymptotically with the refinement of mesh, which led us to use the Richardson's extrapolation given by Equation (86), estimating the apparent order. In the end, Moldex3D® managed to achieve the same accuracy of OpenFOAM® for all fields with M4.

In terms of performance, the proprietary software is clearly faster than the open-source for the same level of refinement. For M1, Moldex3D® was 523 times faster than the OpenFOAM®, for M2 the difference was of 107 times, and for M3 the difference was of 26 times, although, they were not runned in the same number of processors. However, when analysing degrees of refinement with the same magnitude of errors, which means M3 in Moldex3D® and M2 in OpenFOAM®, the proprietary was 10 times faster than the open-source one. However, if we consider M4 in Moldex3D® and M3 in OpenFOAM®, the proprietary software was only four times faster than the open-source one, however the number of the processors used was not the same. For proper comparisons, the same number of processors should be employed. However, as happens in a regular company we have a restricted number of licenses in Moldex3D, which do not allowed us to make that study. In fact, since in OpenFOAM® the only limitation is the computational resources that are available to the user, if we increase the number of processors we will eventually obtain the same performance of the commercial software, as shown in Table 27.

All this results showed that *openInjMoldSim* solver from OpenFOAM® is clearly capable of simulating the filling stage of the injection moulding process, and seems to present a better accuracy than the commerical software Moldex3D®. Although, the open-source one still needs some work on its performance. In order to finish this study of assessment is important to compare this theoretical results with experimental ones to conclude which software gives a better prediction of this phase of the process.

7. Conclusions and Future Work

Injection moulding is the most used polymer processing technique around the world, and present a great complexity of effects that range from heating and melting, to cooling solidification, passing through contraction, distortions, warpage, etc.

The numerical modelling of this process is an area that started to be studied long time ago, and nowadays, the commercial softwares are well mature, presenting several modules employed on daily basis by huge companies to support their development process. However, the commercial softwares present two major drawbacks, which are related to the high cost of the licenses, and also the low flexibility in adapting the code to user specific needs. Therefore, typically the small to medium sized companies cannot withstand the cost of modelling tools. Thus, the search for open-source alternatives as gained a huge interest, and nowadays, some feature of the injection moulding process are already available in open-source softwares. For example, the OpenFOAM® computational library has already capabilities to simulate the filling stage of the injection moulding process assuming a compressible, non-isothermal, multiphase fluid flow. However, there are several improvements which can be done to the currently available codes to accurately predict the physical phenomena of the injection moulding process, and to speed-up the numerical codes. More, the lack of successful case studies is still a drawback to the dissemination of the open-source softwares.

The objective of this work was to validate and assess the *openInjMoldSim* solver, an open-source numerical algorithm available to simulate the injection moulding process in the OpenFOAM® computational library, and the preparation of a successful case study that could be replicated experimentally.

The verification/validation of the open-source solver was performed with three different case studies, where the results obtained with the open-source solver were compared with analytical solutions, and, especially, comparing the results obtained with OpenFOAM® and Moldex3D®, which is a commercial software widely employed in the industry. Furthermore, the simulation of geometries typical of the injection moulding process, which can be assessed experimentally in the future was also one of the main objectives of the current work.

The first case study served to identify the more appropriate modelling setup, and to understand how the numerical algorithm created to model the injection moulding process was built, considering a Newtonian fluid. The results obtained with this first case study allowed to conclude that the artificial interface compression term presented in the VOF formulation gives good results for coarse and medium meshes, however, in refined meshes, it affects mass conservation negatively. Furthermore, the open-source solver presented a good agreement with the analytical solutions. Finally, the incompressible variation did not present substantial differences in relation to the compressible one.

In the second case study, two variations for the equation of state (compressible and incompressible) based on the modified Tait model were used. This study allowed to notice that in the presence of incompressible variations, the performance of the numerical algorithm was improved without a significant loss of precision, which indicates that incompressible formulations are a good approximation for the filling stage of the injection moulding process, and may be a way of increasing solvers efficiency.

In the third case study, comparing the results obtained with OpenFOAM® and Moldex3D®, the first one presented results practically converged, *i.e*, with relative errors below below 10% within three levels of refinement, more, its accuracy was better than the proprietary software for each level of refinement. In order to reach the same order of magnitude of errors, the commercial softwares needed at least four levels of refinement, although it did not present convergence of the errors with mesh refinement. Furthermore, many non-physical results appeared in the commercial software both in terms of field variables contours, and in the time history of inlet pressure, in which, for instance, the passage through the gate could not be identified. The open-source software did not present these limitations. All this results led to the conclusion that OpenFOAM® software presents a better accuracy than the Moldex3D®. In terms of performance, the scenario was reversed with Moldex3D® being clearly faster than the OpenFOAM® solver, however, one of the main limitations of the commercial software is the number of licenses available, therefore, if there is computational resources available, open-source software can match the calculation speed of the proprietary one.

Some issues could not be solved during this thesis, and can be addressed in future works, namely:

- Experimental assessment of the numerical results obtained, especially the ones obtained in the tensile test specimen case study, because the mould is already available in the Department of Polymer Engineering from University of Minho;
- Prepare similar study in other injected moulded parts, typical from industry, and therefore, create more successful case studies;
- Prepare case studies with different materials, and extend this work to semi-crystalline polymers;
- Create a fully incompressible formulation for the filling phase of the injection moulding process, and compare the results obtained with the ones obtained with compressible formulation, and, make a similar study with the commercial software;
- Improve the numerical algorithm employed in the calculations to make the solver

more efficient to relax the requirements of low Courant numbers, because PIMPLE algorithm should cope well with Courant number near the unity;

- Introduce the numerical solution for the packing and cooling phases, and perform a similar comparison with commercial softwares;
- Improve the numerical strategy to deal better with the relevant range of properties typical for the injection moulding process, because many times the problems presented in the numerical calculations was due to the huge difference between the properties of polymer and air;

References

[1] B. Araújo, J. Teixeira, A. Cunha, c. Groth, Parallel three-dimensional simulation of injection molding process, International Journal For Numerical Methods In Fluids, 59(7):801-815(2009).

[2] N. Mole, K. Krebelj, B. Stok. Injection molding simulation with solid semi-crystalline polymer mechanical behavior for ejection analysis. International Journal of Advanced Manufacturing Technology, 93(9-12):4111-4124(2017).

[3] Nagy, Jozsef, Steinbichler, Georg. Fluid Dynamic and Thermal Modeling of the Injection Molding Process in OpenFOAM, in J. Miguel, Nobrega, Jasak, Hrvoje(Eds.), OpenFOAM Selected Papers of the 11th Workshop, Springer, 2019, ISBN 978-3-319-60845-7.

[4] OpenFOAM, The open source CFD toolbox (17 of September of 2019). Retrived https://www.openfoam.com/

[5] K. Krebelj, N. Mole, B. Stok. Three-dimensional modeling of the stress evolution in injection molded parts based on a known melt pressure field. International Journal of Advanced Manufacturing Technology, 90:2363-2376(2017).

[6] K. Krebelj, M. Halilovic, N. Mole. The cooling rate dependence of the specific volume in amorphous plastic injection molding. International Journal of Advanced Manufacturing Technology, 103(1-4):1175-1184(2019).

[7] K. Krebelj and J. Turk, ‘Opensource injection molding simulation. A solver for OpenFOAM’. [Online]. Available: https://github.com/krebeljk/openInjMoldSim. [Accessed: 10-Set-2019].

[8] Moldex 3D (19 of September of 2019). Retrieved from https://www.moldex3d.com/en/.

[9] P. Atkins, Julio de Paula, Physical Chemistry, 9th edition, W. H. Freeman and Company, Oxford, Great Britain, 2010, ISBN 1-4292-1812-6.

[10] James E. Mark, Physical Properties of Polymers Handbook, second edition, Springer, New York, 2007, ISBN 978-0-387-31235-4.

[11] Huamin Zhou, Computer Modelling for Injection Molding, John Wiley & Sons, Inc., New Jersey, 2013, ISBN 978-0-470-60299-7.

[12] P. Kennedy, R. Zheng, Flow Analysis of Injection Molds, second edition, Hanser, Munich, 2013, ISBN 978-1-56990-512-8.

[13] Goodman, Sidney W. Handbook of Thermoset Plastics, 2nd edition, Andrew William, 2000. ISBN 9780815514213.

[14] Kolapo, A. and Olagoke O. Handbook of thermoplastics 2nd edition, CRC Press, 2016. ISBN 978-1-4665-7723-7.

[15] D. V. Rosato, M. G. Rosato. Injection Molding Handbook (Third Edition), Springer, 2000, ISBN 9780792386193.

[16] Piorkowska E. and Rutledge G. Handbook of Polymer Crystallization, John Wiley & Sons, 2013. ISBN 978-0-470-38023-9.

[17] D. W. Van Krevelen, and K. Te Nijenhuis. Properties of Polymers: Their Correlation with Chemical Structure; Their Numerical Estimation and Prediction from Additive Groups Contributions, forth edition, Elsevier, Amesterdan, 2009, ISBN 978-0-08-054819-7.

[18] Crawford R. J., Plastics Engineering 3rd edition, Butterworth-Heinemann, 1998. ISBN 9780750637640.

[19] Predictable Designs (18 of September of 2019). Retrieved from https://predictabledesigns.com/introduction-to-injection-molding/.

[20] L. Xie, L. Shen, B. Jiang, Modelling and Simulation for Micro Injection Molding Process, in:Igor Minin (Ed.), Computational Fluid Dynamics Technologies and Applications, InTech, 2011. ISBN: 978-953- 307-169-5.

[21] Vanessa Goodship, ARBURG Pratical Guide to Injection Moulding, first edition, Rapra Technology Limited, 2010, ISBN 1-85957-444-0.

[22] V.Itha, M.Irfaan, V.Reddy, Automatic Injection Moulding by using PLC's, International Journal of Science, Engineering and Technology. 3(2):423-427 (2015).

[23] Piau J-M., and Agassant J-F. Rheology for Polymer Melt Processing, Elsevier, 1996. ISBN 0-444-82236-4.

[24] John D. Ferry, Viscoelastic Properties of Polymers, third edition, John Wiley & Sons, New York, 1980, ISBN 0471048941.

[25] Kwang S. C. Viscoelasticity of Polymers. Springer Netherlands, 2016. ISBN: 978-94-017-7562-5.

[26] Gerhard A. Holzapfel, Nonlinear Solid Machanics: A Continuum Approach for Engineering, first edition, John Wiley & Sons Ltd, Baffins Lane, Chichester England, 2000, ISBN 0 471 82304 X (ppc)

[27] André Berger, Numerical Modelling of Composite Materials Based on a Combined Manufacturing-Crash Simulation,PhD Thesis, Queen Mary University of London, 2014.

[28] R. Bird, R. Armstrong, O. Hassager, Dynamics of Polymeric Liquids, second edition, John Wiley & Sons, Canada, 1987, ISBN 047180245X.

[29] Agassant, Jean-François; Avenas, Pierre; Carreau, Pierre J. Polymer Processing 2 edition: Principles and Modeling, Hanser, Munich, 2017, ISBN 978-1-56990-605-7.

[30] Polymer properties database, https://polymerdatabase.com/polymer%20physics/Viscosity2.html, 17-11-2019.

[31] Ostwald W. About the rate function of the viscosity of dispersed systems, Kolloid Z, 36:99-117(1925).

[32] De Waele A. Viscometry and plastometry, Oil Color Chem Assoc J, 6:33-88(1923).

[33] Gottfert, Rheo-info: Modelling Viscosity data, 2018.

[34] Carreau P. Ph.D Thesis, University of Wisconsin, Maddison, 1968.

[35] Yasuda K. Ph. D.. Thesis, Massachusetts Institute of Technology, Cambridge, 1979.

[36] M.M. Cross. Rheology of non-Newtonian fluids: A New flow equation for pseudo-plastic systems. J. Colloid Sci., 20:417–437, 1965.

[37] C. Hieber, H. Chiang, Shear-Rate-Dependence Modelling of Polymer Melt Viscosity, Polymer Engineering and Science, 32(14):931-938(1992).

[38] Richards, W. D., and D. K. Yoshimura. A Viscosity Model for Linear Polycarbonate Resins. ANTEC 90. Plastics in the Environment: Yesterday, Today & Tomorrow (1990): 1603-1606.

[39] R. Tanner, K. Walters, Rheology: An Historical Perspective, Rheology Series, Elsevier, 1980.

[40] Williams M. L., Landel R. F. and Ferry, J. D. The Temperature Dependence of Relaxation Mechanisms in Amorphous Polymers and Other Glass-forming Liquids, Journal of the American Chemical Society, 77:3701-3707(1955).

[41] Mavridis M, Shroff RN. Temperature dependence of polyolefin melt rheology, Polym Eng Sci, 32:1778-1791(1992).

[42] Fernandez M, Munoz ME, Santamaria A, Syrjälä S, Aho J. Determining the pressure dependency of the viscosity using PVT data: A practical alternative for thermoplastics, Polym Testing, 28:109-113(2009).

[43] Omnexus, the material selection platform, https://omnexus.specialchem.com/polymer-properties/properties/glass-transition-temperature, 11-01-2020.

[44] O. S. Carneiro, J. M. Nóbrega. Design of Extrusion Forming Tools, iSmithers Rapra Publishing, 2013. ISBN 978-1-84735-517-1.

[45] Goubert A, Vermant J, Moldenaers P, Göttfert A, and Ernst B. Comparison of measurement techniques for evaluating the pressure dependence of the viscosity. Applied Rheo, 11:26-37(2001).

[46] I. L.Ngo, S. Jeon, C. Byon. Thermal conductivity of transparent and flexible polymers containing fillers: A literature review. International of Heat and Mass Transfer. 98:219-226(2016).

[47] Ress, H. Understanding Injection Mold Design, Hanser Publications, 2001. ISBN 978-1-59124-795-1.

[48] Wang, J. PVT Properties of Polymers for Injection Molding. In Some Critical Issues for Injection Molding, J. Wang, Ed. 2012.

[49] Tait PG, Physics and Chemistry of the Voyage of the HMS Challenger, Vol. 2, Part 4, HMSO, London, 1888.

[50] B. S. Chen and W. H. Liu, Numerical Simulation of the Post-Filling Stage in Injection Molding with a Two-Phase Model, Polymer Engineering and Science. 34(10):835-846(1994).

[51] J. Wang et al. Modeling of pvT behavior of semi-crystalline polymer based on the two-domain Tait equation of state for injection molding, Materials and Design. 183:108-149(2019).

[52] I. I. Rubin, Injection Molding Theory and Practice, John Wiley & Sons, New York, 1972.

[53] D. Cardozo, Three Models of the 3D Filling Simulation for Injection Molding: A Brief Review, Journal of Reinforced Plastics and Composites, 27(18):1963-1974(2008).

[54] G. Titomanlio, V. Speranza, and V. Brucato. On the simulation of thermoplastic injection moulding process, Part 2: Relevance of interaction between flow and crystallization. Int. Polym. Process, 12:45-53,1997.

[55] H.L. Toor, R.L. Ballman, and L.Cooper. Predicting mold flow by eletronic computer., Modern Plastics, December, 1960.

[56] Gilmore, G. D. and Spencer, R. S. Role of Pressure, Temperature, and Time in Injection Molding Process, Modern Plastics, 37(8):143-151(1950).

[57] Spencer, R. S. and Gilmore, G. D. Residuals Strains in Injection Molded Polystyrene, Modern Plastics, 28(4):97-105(1950).

[58] I.T. Barrie. Understanding how an injection mold fills. SPE Journal, 27:64-69, 1971.

[59] R. L. Ballman, H. L. Toor, Injection Molding: Rheological Interpretation, Modern Plastics, 37(1-2):105-111(1950).

[60] J. Stevenson, A Simplified Method for Analysing Mold Filling Dynamics. Part I: Theory, Polymer Engineering & Science. 18(7):577-582(1978).

[61] R. Nunn, R. Fenner, Flow and Heat Transfer in the Nozzle of an Injection Molding Machine, Polymer Engineering & Science. 17(11):811-818(1977).

[62] M.R. Kamal, and S. Kenig. The injection moldfing of thermoplastics, Part II: Experimental test of the model. Polymer Engineering Science, 12:302-308, 1972.

[63] G. Williams and H.A. Lord. Mold filling studies for the injection molding of thermoplastic materials. Part I: The flow of plastic materials in hot and cold walled circular channels. Polymer Engineering and Science, 15:553-568, 1975.

[64] H. Lord, G. Williams, Mold-Filling Studies for the Injection Molding of Thermoplastic Materials. Part II: The Transient Flow of Plastic Materials in the Cavities of Injection-Molding Dies, Polymer ngineering & Science. 15(8):569-582(1975).

[65] C.A. Austin. Moldflow Design Principles. Moldflow Pty. Ltd, Melbourne, Australia, 1985.

[66] Autodesk Moldflow (17 of September of 2019). Retrieved from https://www.autodesk.com/products/moldflow/overview.

[67] C. Austin, Filling of mold cavities, In E. C. Bernhardt, editor, Computer-Aided-Engineering for Injection Molding, Hanser, New York, 1983.

[68] M. Kamal. The McGill University model(McKam-II):A comprehensive, integrated computer simulation of the injection molding process. In L.T.Manzione, editor, Application of Computer-Aided Engineering in Injection Molding. Hanser, New York, 1987.

[69] P. Lafleur, M. Kamal, A Structure-Oriented Computer Simulation of the Injection Molding of Viscoelastic Crystalline Polymers Part I: Model with Fountain Flow, Packing, Solidification, Polymer Engineering & Science. 26(1):92-102(1986).

[70] C. Hieber, S. Shen, A finite-element/finite-difference of the injection-molding filling process, Journal of Non-Newtonian Fluid Mechanics, 7:1-32(1980).

[71] Hele-Shaw, H. S. The flow of water. Nature (London), 58:34(1898).

[72] Hieber, C. A. and Shen S. F. Flow analysis of the nonisothermal 2-dimensional filling process in injection-molding., Israel Journal of Technology, 16(5-6):248-254(1978).

[73] Hieber C. A., Shen S. F. A finite-element/finite-difference simulation of the injection-molding filling process., Journal of Non-Newtonian Fluid Mechanics, 7(1):64-69(1980).

[74] C. W. M. Sitters. Numerical Solution of Injection Moulding. PhD Thesis, Technical University of Eindhoven, 1988.

[75] A.H.M Boshouwers and J. J. Van der Werf. A Simulation Code for the Filling Phase of Injection Moulding Process of Thermoplastics. PhD thesis, Technical University of Eindhoven, 1988.

[76] A. I. Isayev. Orientation, residual stresses and volumetric effects. In A. I. Isayev, editor, Injection and Compression Fundamentals, pages 227-328. Marcel Dekker Inc., New York, 1987.

[77] T. H. Kwon, S. F. Shen, K. K. Wang. Computer-aided cooling system design for injection molding. In Soc. Plast. Eng. Proc. Annual Tech. Conf. (ANTEC),32:110-115, 1986.

[78] K. Himasekhar, C.A. Hieber, and K.K. Wang. Computer-aided design software for cooling system in injection molding, In Soc. Plast. Eng. Proc. Annual Tech. Conf. (ANTEC), 35:352-355(1989).

[79] T. E. Burton and M. Rezayat. POLYCOOL 2: A three-dimensional transient mold-cooling simulator. In L: T. Manzione, editor, Applications of Computer Aided Engineering in Injection Molding. Hanser, New York, 1987.

[80] A. I. Isayev, C. A. Hieber and D. L. Crouthamel. Residual stressses in injection molding of amorphous polymers. In Soc. Plast. Eng. Proc. Annual Tech. Conf. (ANTEC), 27:110-113, 1981.

[81] G. Titomanlio, V. Brucato and M. R. Kamal. Mechanism of cooling stress build-up in injection molding of thermoplastic polymers. Int. Polym. Process, 1:55-59, 1987.

[82] N. Santhanam and K.K. Wang. A theoretical and experimental investigation of warpage in injection molding, In Soc. Plast. Eng. Proc. Annual Tech. Conf. (ANTEC), 36:270–273(1990).

[83] H. H. Chiang, C. A. Hieber, K. K. Wang. A unified simulation of the filling and post filling stages in injection molding. Part I: Formulation and Part II: Experimental verification. Polym. Eng. Science. 31:116-139,1991.

[84] L. F. Douven. Towards the Computation of Properties of Injection-Moulded Products: Flow and Thermally induced stresses in Amorphous Thermoplastics. PhD Thesis, Technical University of Eindhoven, 1991.

[85] F. Baaijens, Calculation of residual stresses in injection molded products, Rheological Acta. 30(3):284-299(1991).

[86] L. Caspers. An integrated Approach to the Simulation of Injection Moulding. PhD Thesis, Technical University of Eindhoven, 1995.

[87] G. Titomanlio, V. Speranza, On the Simulation of Thermoplastic Injection Moulding Process, International Polymer Processing.12(1):45-53(1995).

[88] J. F. Hétu, Y. Lauzé, A. Garcia-Rejon. Three-dimensional finite element simulation of mold filling processes. In S. F. Shen and P. Dawson, editors, Siulation of Materials Processing: Theory, Methods and Applications, Numinform 95, Rotterdam, 1995. Balkema.

[89] C. Friedl. Progress towards true 3D analysis for injection molding. In Soc. Plast. Eng. Proc. Annual Tech. Conf. (ANTEC), volume 42, 1996.

[90] V. Rajupalem, K. Talwar, and C. Friedl. Three-dimensional simulation of the injection molding process. In Soc. Plast. Eng. Proc. AnnualTech. Conf. (ANTEC), 43:670–673, 1997.

[91] K. Talwar, F. S. Costa, V. Rajupalem, L. Antanovski, and C. Friedl. Three-dimensional simulation of plastic injection molding. In Soc. Plast. Eng. Proc. Annual Tech. Conf. (ANTEC), 46:563-566, 1998.

[92] R. Chang, W. Yang, Numerical simulation of mold filling in injection molding using a three-dimensional finite volume approach, International Journal For Numerical Methods In Fluids, 37(2):125-148(2001).

[93] S. V. Patankar. Numerical heat transfer and fluid flow., Hemisphere Publishing Corporation, 1981.

[94] Ferziger J. H. and Perić M. Computational Methods for Fluid Dynamics, third edition, Springer, 2002. ISBN 3-540-42074-6.

[95] G. Eder, H. Janeschitz-Kriegl. Crystallization. In H. E. H. Meijer, editor, Materials, Science and Technology, Vol. 18, Processing of Polymers, 269-342. Wiley-VCH, New York, 1997.

[96] G. Eder, H. Janeschitz-Kriegl, and S. Liedauer. Crystallization processes in quiescient and moving polymer melts under heat transfer conditions.Prog. Polym. Sci., 15:629-714, 1990.

[97] E. Koscher, R. Fulchiron, Influence of shear on polypropylene crystallization: morphology development and kinetics, Polymer. 43(25):6931-6942(2002).

[98] H. Zuidema. Flow-Induced Crystallinity of Polymers, Applications to Injection Molding. PhD thesis , Technical University of Eindhoven, 2000.

[99] W. Schneider, A. Koppl, and J. Berger. Non-isothermal crystallizaion, Crystallization in Polymers. Int. Polym. Process., 2:151-154, 1988.

[100] R. Pantani, I. Coccorullo, V. Speranza, G. Titomanlio, Modeling of morphology evolution in the injection molding process of thermoplastic polymers, Progress in Polymer Science. 30(12):1185-1222(2005).

[101] Y. Kwon, E. Lim, Y. Song, Simulation of injection-compression molding for thin and large battery housing, Current Applied Physics. 18(11):1451-1457(2018).

[102] Q. Jiang, H. Liu, Q. Xiao, S. Chou, A. Xiong, H. Nie, Three-dimensional numerical simulation of total warpage deformation for short-glass-fiber-reinforced polypropylene composite injection-molded parts using coupled FEM, Journal of Polymer Engineering. 38(5):493-502(2018).

[103] G. Shen, Q. Xiong, P. Morris, E. Paterson, A. Sergeev, Y. Wang. OpenFOAM for Computational Fluid Dynamics, Notices of the American Mathematical Society, 61(4):354-363(2014).

[104] Habla, Florian, Fernandes, C., Maier, Maximilian, Densky, Lennart, Ferrás, L. L., Rajkumar, Ananth, Carneiro, O. S., Hinrichsen, Olaf, Nóbrega, J. Miguel. Development and validation of a model for the temperature distribution in the extrusion calibration stage, Applied Thermal Engineering. 100:538-552(2016).

[105] Rajkumar, A., Ferrás, L. L., Fernandes, C., Carneiro, Olga S., Nóbrega, J. Miguel. Guidelines for balancing the flow in extrusion dies: The influence of the material rheology, ournal of Polymer Engineering, 38:(2):197-211(2017).

[106] A. Magalhães. OpenFOAM simulation of the injection moulding filling stage, Master Thesis, University of Minho, 2016.

[107] A. Margarida. Injection moulding simulation using OpenFOAM, Master Thesis, University of Minho, 2019.

[108] H. K. Versteeg, W. Malalasekera, An Introduction to compuattional Fluid Dynamics: The Finite Volume Method. Pearson Education Limited, Second edition. ISBN: 978-0-13-127498-3.

[109] J.Hétu, D. Gao, A. Garcia-Rejon, G. Salloum, 3D Finite Element Method for the Simulation of the Filling Stage in Injection Molding, Polymer Engineering and Science. 38(2):223-236(1998).

[110] S. Kim, L. Turng, Developments of three dimensional computer-aided engineering simulation for injection moulding, Modelling and Simulation in Materials Science and Engineering. 12(3):151-173(2004).

[111] C. Fetecau, I. Postolache, F. Stan, Numerical and Experimental Study on the Injection Molding of a Thin-Wall Complex Part, in Proccedings of the International Manufacturing Science and Engineering Conference (MSEC). 1:85-93(2008).

[112] P. Guerrier, G. Tosello, J. H. Hattel, Flow visualization and simulation of the filling process during injection molding, Journal of Manufacturing Science and Technology, 16:12-20(2017).

[113] M. Tutar, A. Karakus, Injection molding simulation of a compressible polymer, Journal of Polymer Engineering, 29(6):355-384(2009).

[114] C. Fernandes, António J. Pontes, Júlio César Viana, António Gaspar-Cunha, Modelling and Optimization of the Injection-Molding Process: A review, Advances in Polymer Technology. 37(2):429-449(2016).

[115] J. Liang, W. Luo, Z. Huang, H. Zhou, Y. Zhang, Y. Zhang, Y. Fu A robust finite volume method for three-dimensional filling simulation of plastic injection molding, Engineering Computations. 34(3):814-831(2016).

[116] K. Choi, B. Koo, Analysis of the three dimensional mold-filling process in injection molding, in proceedings of SPE - Annual Technical Conference. 1(3):501-505(2004).

[117] H. Zhou, T. Geng, D.Li, Numerical Filling Simulation of Injection Molding Based on 3D Finite Element Model, Journal of Reinforced Plastics and Composites. 24(8):823-830(2005).

[118] F. Ilinca, J. Hétu, 3-D simulation of the packing-cooling stage in polymer injection molding, in processing of SPE - Annual Technical Conference. 1(2):5(2000).

[119] Himasekhar K. Numerical simulation of mold heat transfer of injection molded plastic parts using a modified three-dimensional boundary element method. International Communications in Heat and Mass Transfer, 16(1):55-64(1989).

[120] Kwon, T. H. Mold Cooling System Design Using Boundary Element Method., Journal of Engineering for Industry, 110(4):384-394(1988).

[121] Heiss, J. F. and Coull, J. Nomograph of Dittus-Boelter Equation for Heating and Cooling Liquids., Industrial & Engineering Chemistry, 43(5):1226-1229(1951).

[122] Rusche, Henrik, Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions, Ph.D Thesis submitted to Department of Mechanical Engineering from the Imperial College of Science, Technology and Medicine, 2002.

[123] Jasak, Hrvoje, Error Analysis and Estimation for the Finite Voulme Method with Applications to Fluid Flow, Ph.D Thesis submitted to Department of Mechanical Engineering from the Imperial College of Science, Technology and Medicine, 1996.

[124] C. Hirsch. Numerical computation of internal and external flows, John Wiley & Sons, 1991.

[125] Ubbink, Onno, Numerical Prediction of two fluid systems with sharp interfaces. Ph.D. Imperial College of Science, Technology & Medicine.

[126] Darwish, M.; Mangani, L.; Moukalled, F The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab, Springer, 2016. ISBN 978-3-319-16873-9.

[127] Antonio M., Paulo L., Guilherme C., Luiz S. Analysis of the Non-Orthogonality Correction of Finite Volume Discretization on unstructured Meshes., Conference: 22nd International Congress of Mechanical Engineering (COBEM 2013), 2013.

[128] Tobias Holzmann. Mathematics, Numerics, Derivations and OpenFOAM, Holzmann CFD, URL www.holzmann-cfd.de, DOI: 10.13140/RG.2.2.27193.36960.

[129] S. V. Patankar, and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. International Journal Heat Mass Transfer, 15:1787(1972).

[130] R. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational Physics. 62:40-65,(1986).

[131] Hirt C. W. and Nicols B. D. Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comp. Phys., 39:201-225(1981).

[132] B. Celik Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications, Journal of Fluids Engineering. 130, 1-4 (2008).

[133] Oberkampf, W. L., Trucano, T. G. and Hirsch, C. Verification, validation, and predictive capability in computational engineering and physics, Applied Mechanics Reviews, 57(5):345-384(2004).

[134] Celik, I., Karatekin, O. Numerical experiments on application of Richardson Extrpolation with nonuniform grids., Journal of Fluids Engineering, 119(3):584-590(1997).

[135] Pimenta, F. and Alves, M. A. Stabilization of an open-source finite-volume solver for viscoelastic fluid flows, Journal of Non-Newtonian Fluid Mechanics. 239:85-104(2017).

[136] cfMesh, A Library for Automatic Mesh Generation, Creative Fields, Dr. Franko Juretic, https://cfmesh.com/about/

[137] Christopher J. Greenshields, OpenFOAM User Guide Version 7, the OpenFOAM Foundation, 2019. https://openfoam.org

[138] K. H. Kim, A. I. Isayev, K. Kwon, and C. van Sweden. Modelling and experimental study of birefringence in injection molding of semicrystalline polymers. Polymer, Elsevier. 46(12):4183-4203, 2005.

[139] Richardson L. F. The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, With an Application to the Stresses in a Masonary Dam, Philos. Trans. R. Soc. London, Ser. A, 210:307-357(1910).

[140] Z. Tadmor and C. Gogos, Principles in Polymer Processing, second edition, John Wiley & Sons, New Jersey, 2006. ISBN: 9780471387701.

Appendices

Appendix A. Set of invariants of the rate of deformation tensor

Recalling equation for the rate of deformation (7), it has the following form

$$\underline{\underline{D}} = \frac{1}{2}\left(\nabla \underline{u} + (\nabla \underline{u})^{\mathrm{T}}\right). \tag{88}$$

This equation can be written in his extensive form like

$$\underline{\underline{D}} = \frac{1}{2}\left(\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} + \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{bmatrix}\right), \tag{89}$$

And simplifying it

$$\underline{\underline{D}} = \frac{1}{2}\begin{bmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & 2\frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} & \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} & 2\frac{\partial w}{\partial z} \end{bmatrix}. \tag{90}$$

Equation (90), shows that the rate of deformation tensor is a symmetric tensor. For any symmetric tensor, there is a unique frame of reference in which the tensor is a diagonal matrix [29], like

$$\underline{\underline{M}} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{91}$$

λ_i are the eigenvalues or principal values, and the frame of reference is defined by the principal directions in fluid mechanics. The eigenvalues are characteristics of a tensor and independent of the choice of the coordinate system [29]. They are the solution of the characteristic equation

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0 \tag{92}$$

where

$$I_1 = \lambda_1 + \lambda_2 + \lambda_3 = trM, \tag{93}$$

$$I_2 = \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1, \tag{94}$$

$$I_3 = \lambda_1\lambda_2\lambda_3, \tag{95}$$

I_1, I_2, I_3 are the invariants of the tensor $\underline{\underline{M}}$. A different set of invariants can be defined for $\underline{\underline{M}}$

[29], like

$$J_1 = m_{11}+m_{22}+m_{33} = tr\underline{\underline{M}}, \tag{96}$$
$$J_2 = \frac{1}{2}\sum_{ij} m_{ij}^2 = \frac{1}{2}tr\underline{\underline{M}}^2, \tag{97}$$
$$J_3 = \frac{1}{3}\sum_{ijk} m_{ij}m_{jk}m_{ki} = \frac{1}{3}tr\underline{\underline{M}}^3, \tag{98}$$

This second group of invariants are in fact a combination of the first three, and they are related by

$$J_1 = I_1, \tag{99}$$
$$2J_2 = I_1^2 - 2I_2, \tag{100}$$
$$3J_3 = 3I_3 - 3I_1I_2 + I_1^3. \tag{101}$$

First Invariant of $\underline{\underline{\mathbf{D}}}$

From the explanation given above, the first invariant of the rate of deformation tensor is given by the sum of the diagonal members of its matrix, which means

$$tr\underline{\underline{D}} = \frac{1}{2}\left(2\frac{\partial u}{\partial x} + 2\frac{\partial v}{\partial y} + 2\frac{\partial w}{\partial z}\right) = \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) = \underline{\nabla}\cdot\underline{u}. \tag{102}$$

For a incompressible fluid which means that density or specific volume are constants, $\rho = \frac{1}{\hat{V}} =$ constant, the continuity equation (37) is given by

$$\frac{\partial\rho}{\partial t} + \underline{\nabla}\cdot(\rho\underline{u}) = 0 \Leftrightarrow \frac{\partial\rho}{\partial t} + \rho\underline{\nabla}\cdot\underline{u} + \underline{u}\cdot\underline{\nabla}\rho = 0, \tag{103}$$

as ρ is constant, both first and third term of the last equation disappear, and rearranging the terms, we obtain

$$\underline{\nabla}\cdot\underline{u} = 0, \tag{104}$$

And consequently, Equation (102) is zero, which means that for incompressible fluids, the first invariant of $\underline{\underline{D}}$ is zero, and therefore, the set of invariants I have the same absolute value of the set of invariants J.

Second Invariant of $\underline{\underline{\mathbf{D}}}$

In order to understand the second invariant of the rate of deformation tensor, the double dot product or scalar product of a tensor is introduced. For two tensors $\underline{A}$ and $\underline{B}$, the double dot product between both is given by [**?**]

$$\underline{\underline{A}} : \underline{\underline{B}} = A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31} + \tag{105}$$

$$A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32} + \tag{106}$$

$$A_{31}B_{13} + A_{33}B_{33} + A_{32}B_{23} \tag{107}$$

$$= \sum_{i,j}^{3} A_{ij}B_{ji} \tag{108}$$

The second invariant of $\underline{\underline{D}}$ (J_2) is given by Equation (97), and applying the concept of double dot operation between two symmetric tensors given by Equation (108) [**?**], we obtain

$$J_2 = \frac{1}{2}\sum_{ij} D_{ij}^2 = \frac{1}{2}\underline{\underline{D}} : \underline{\underline{D}} = \frac{1}{2} tr \underline{\underline{D}}^2. \tag{109}$$

Since J_2 is a measure of the scalar intensity of the tensor, the generalized strain rate of which accounts for the magnitude of the shear-rate can be defined as a function of J_2 [12], given by

$$\dot{\gamma} = \sqrt{2J_{2D}} = \sqrt{2\underline{D} : \underline{D}} = \sqrt{2\sum_{i,j} D_{ij}^2} = \sqrt{2tr\left(\underline{\underline{D}}^2\right)}. \tag{110}$$

Third Invariant of $\underline{\underline{D}}$

From Equation (101), and remembering the fact that the first invariant is zero for incompressible fluids, Equation (101) becomes

$$I_3 = J_3 \tag{111}$$

I_3 and J_3 for a shear dominant flows [28, 29].

As stated in this appendix, one can understand that in injection moulding process, the first and third invariants of the rate of deformation tensor ($\underline{\underline{D}}$) are zero. Since it is not possible to make an explicit dependence of η on $\underline{\underline{D}}$, as explained in Chapter 2.3, the relation is made between the viscosity η, and the second invariant of $\underline{\underline{D}}$.

Appendix B. Generalized form of the linear momentum equation

Many authors and many reviews [90, 109, 111], and many other usually refer the conservation of momentum equation (Equation (43)) in his shortened version, meaning in the vectorial form, although is important to notice that the linear momentum conservation equation are three equations given by

$$\text{x direction}: \frac{\partial \rho \underline{u}}{\partial t} + \nabla \cdot (\rho \underline{u}\,\underline{u}) = \frac{\partial\left(-p+\tau_{xx}\right)}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{Mx}, \tag{112}$$

$$\text{y direction}: \frac{\partial \rho \underline{u}}{\partial t} + \nabla \cdot (\rho \underline{u}\,\underline{u}) = \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial\left(-p+\tau_{yy}\right)}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + S_{My}, \tag{113}$$

$$\text{z direction}: \frac{\partial \rho \underline{u}}{\partial t} + \nabla \cdot (\rho \underline{u}\,\underline{u}) = \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial\left(-p+\tau_{zz}\right)}{\partial z} + S_{Mz}. \tag{114}$$

Furthermore, considering an arbitrary property the total derivative of ϕ is given by

$$\rho\left(\frac{D\phi}{Dt}\right) = \rho\left(\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \underline{u})\right). \tag{115}$$

This relation means that the rate change of ϕ per unit of mass (left hand-side of the equation) is equal the sum of the rate change of ϕ in the element volume and the balance of the flux of ϕ in the element of fluid (right hand-side of the equation).

Applying the definition of total derivative to the expanded version of the linear momentum Equations [(112)-(114)] we obtain

$$\text{x direction}: \rho\frac{Du}{Dt} = \frac{\partial\left(-p+\tau_{xx}\right)}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{Mx}, \tag{116}$$

$$\text{y direction}: \rho\frac{Dv}{Dt} = \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial\left(-p+\tau_{yy}\right)}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + S_{My}, \tag{117}$$

$$\text{z direction}: \rho\frac{Dw}{Dt} = \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial\left(-p+\tau_{zz}\right)}{\partial z} + S_{Mz}. \tag{118}$$

Appendix C. Double dot operation between the stress tensor and the velocity gradient

The viscous dissipation term presented in energy conservation equation (Equation (39) is given by the double dot operation between between the stress tensor $\underline{\underline{\sigma}}$ and the velocity gradient $\nabla\underline{u}$ which is given by

$$
\begin{aligned}
\underline{\underline{\sigma}} : \nabla\underline{u} &= \left[-p\underline{\underline{I}} + 2\eta\left(\dot{\gamma},T,p\right)\underline{\underline{D}}\right] : \nabla\underline{u} \\
&= -p\underline{\underline{I}} : \nabla\underline{u} + 2\eta\left(\dot{\gamma},T,p\right)\underline{\underline{D}} : \nabla\underline{u} \\
&= -p\nabla\cdot\underline{u} + \eta\left(\dot{\gamma},T,p\right)\left[\nabla\underline{u} + \left(\nabla\underline{u}\right)^{\mathrm{T}}\right] : \nabla\underline{u} \\
&= -p\nabla\cdot\underline{u} + \frac{1}{2}\eta\left(\dot{\gamma},T,p\right)\left[\nabla\underline{u} + \left(\nabla\underline{u}\right)^{\mathrm{T}}\right] : \left[\nabla\underline{u} + \left(\nabla\underline{u}\right)^{\mathrm{T}}\right] \qquad (119) \\
&= -p\nabla\cdot\underline{u} + 2\eta\left(\dot{\gamma},T,p\right)\underline{\underline{D}} : \underline{\underline{D}} \\
\text{Remeber that } \dot{\gamma} &= \sqrt{2\underline{\underline{D}} : \underline{\underline{D}}} \\
&= -p\nabla\cdot\underline{u} + \eta\dot{\gamma}^2
\end{aligned}
$$

Substituting this relation in the energy conservation equation, we obtain an explicit equation for the temperature that gives the effect the shear-rate has on melt temperature, which means the viscous dissipation promoted by the shear-rate inside mould cavity.

Appendix D. 2D Technical drawing of cylindrical cavity

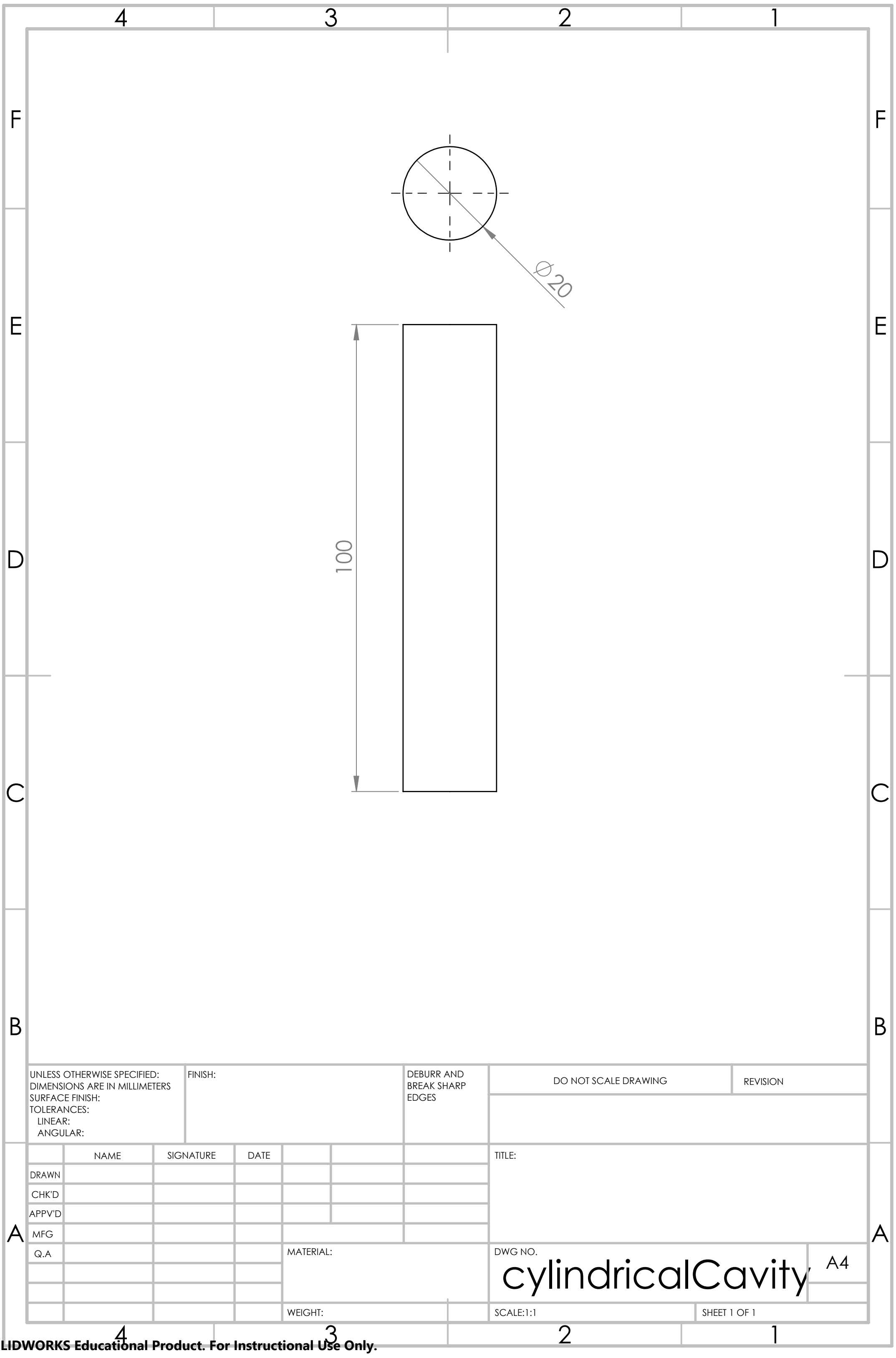

Figure 42: 2D drawing of the cylindrical cavity.

Appendix E. Material Models and Coefficients

The coefficients of the Cross-WLF model for the variation of viscosity with shear-rate, temperature, and pressure used in the GPPS - General purpose polystyrene Styron 678 from americas styrenics are presented in Table 28.

Table 28: Cross-WLF coefficients for both GPPS - Styron 678 from Americas Styrenics.

Parameters	GPPS- Styron 678 Americas Styrenics
n	0.2903
τ^*[Pa]	13 678
D_1[Pa.s]	7.44E+10
D_2[K]	373.15
D_3[K/Pa]	0
A_1	25.971
A_2[K]	51.6
c_p[J/ (kg K)]	2100
k[W/ (m K)]	0.15

The coefficients for the two different variations of modified Tait Model for the equation of state are given in Table 29.

Table 29: Modified Tait model coefficients for three different cases

Parameters	(a) Compressible variation	(b) Incompressible variation
b_{1m} [m^3/kg]	0.0009881	0.001075
b_{2m} [m^3/(kg K)]	7.03E-07	1.00E-09
b_{3m} [Pa]	1.71E+08	5.00E+10
b_{4m} [1/K]	0.004495	1.00E-07
b_{1s} [m^3/kg]	0.0009873	0.001075
b_{2s} [m^3/(kg K)]	2.89E-07	1.00E-09
b_{3s} [Pa]	2.43E+08	5.00E+10
b_{4s} [1/K]	0.003106	1.00E-07
b_5 [K]	373.98	0
b_6 [K/Pa]	2.88E-07	0
b_7 [m^3/kg]	0	0
b_8 [1/K]	0	0
b_9 [1/Pa]	0	0

Appendix F. 2D Technical drawing of rectangular Cavity

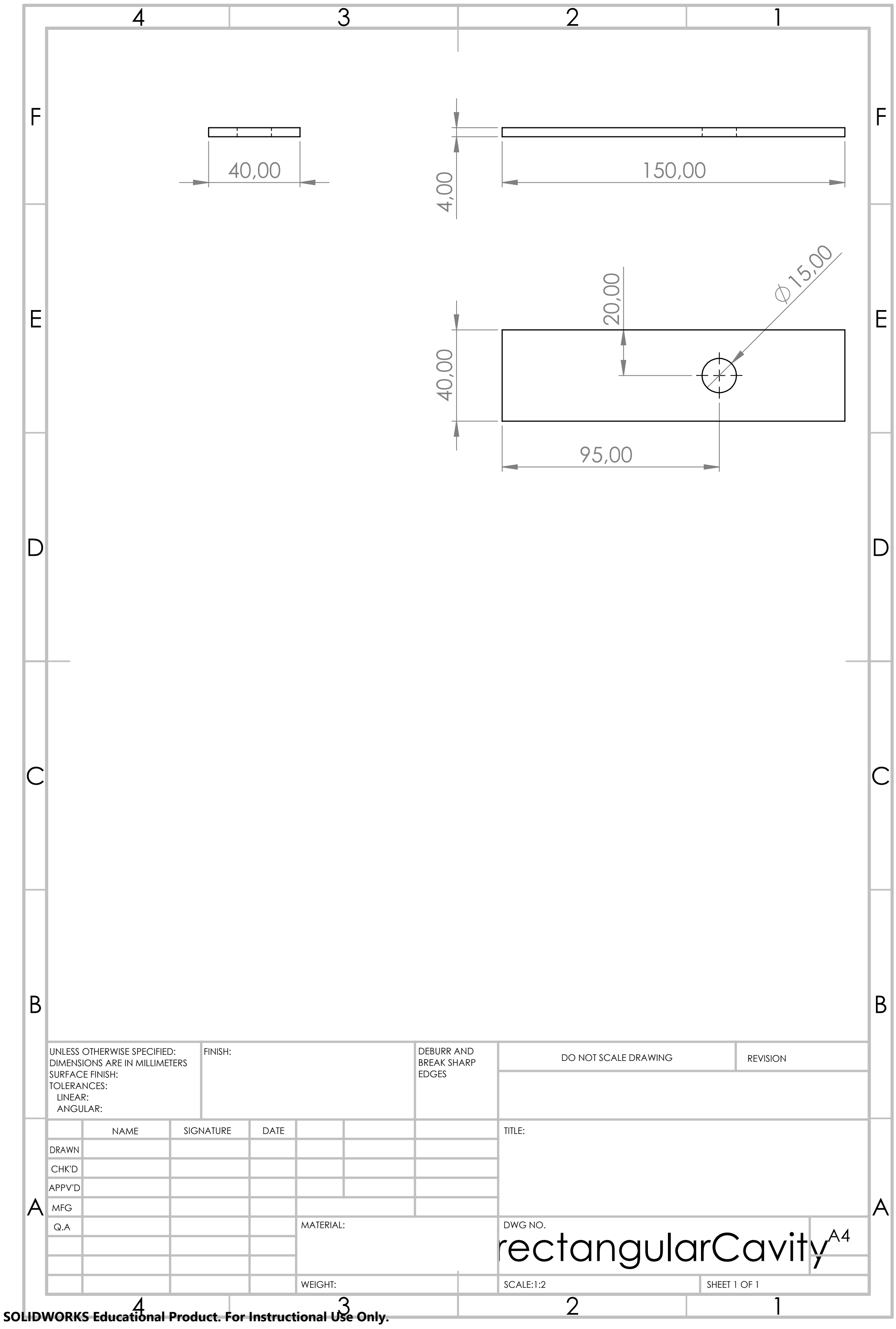

Figure 43: 2D drawing of the rectangular cavity.

Appendix G. 2D Technical drawing of tensile test specimen

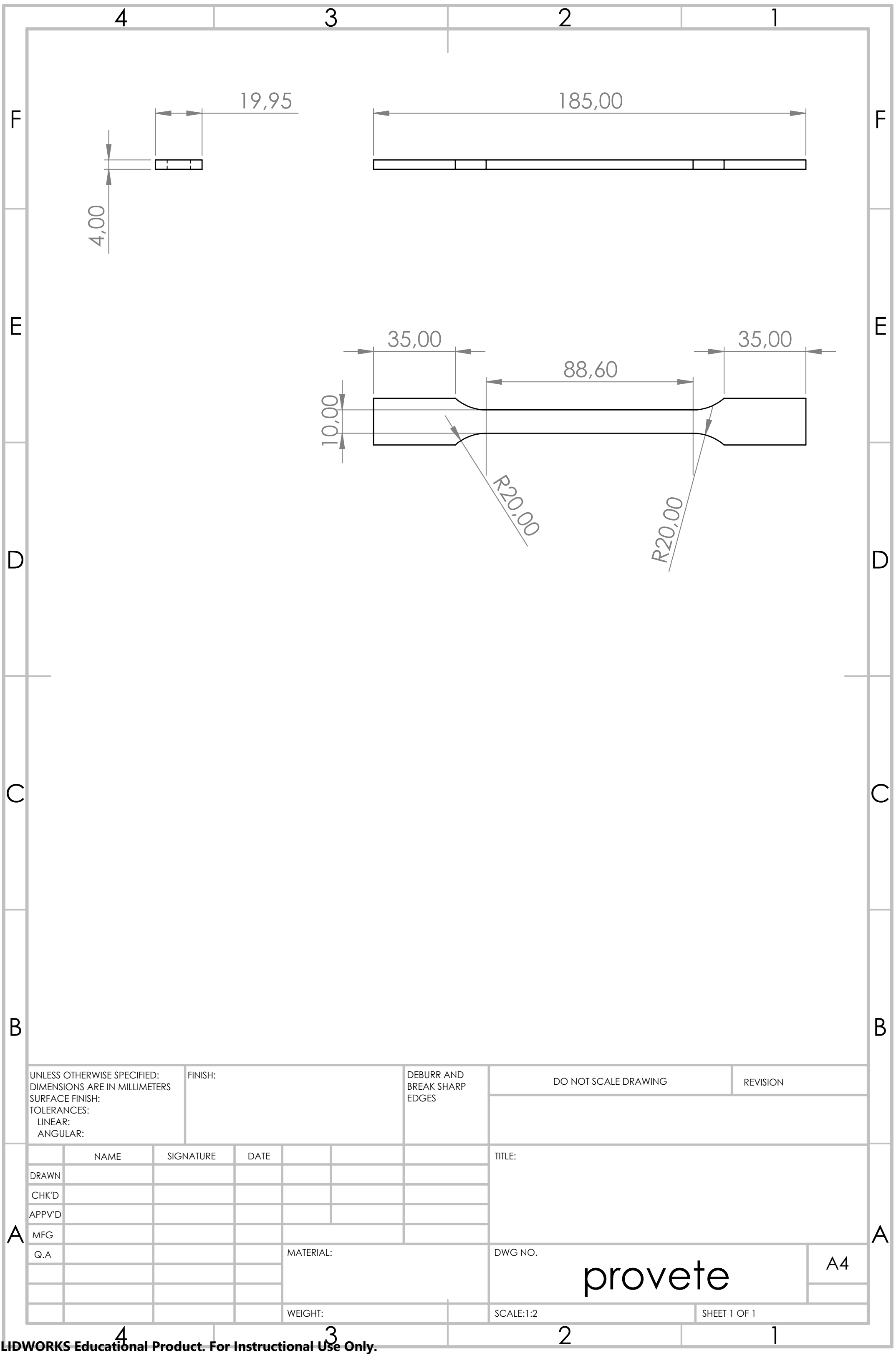

Figure 44: 2D Drawing of the tensile test specimen.

Appendix H. 2D Technical drawing of the feeding system of the tensile test specimen

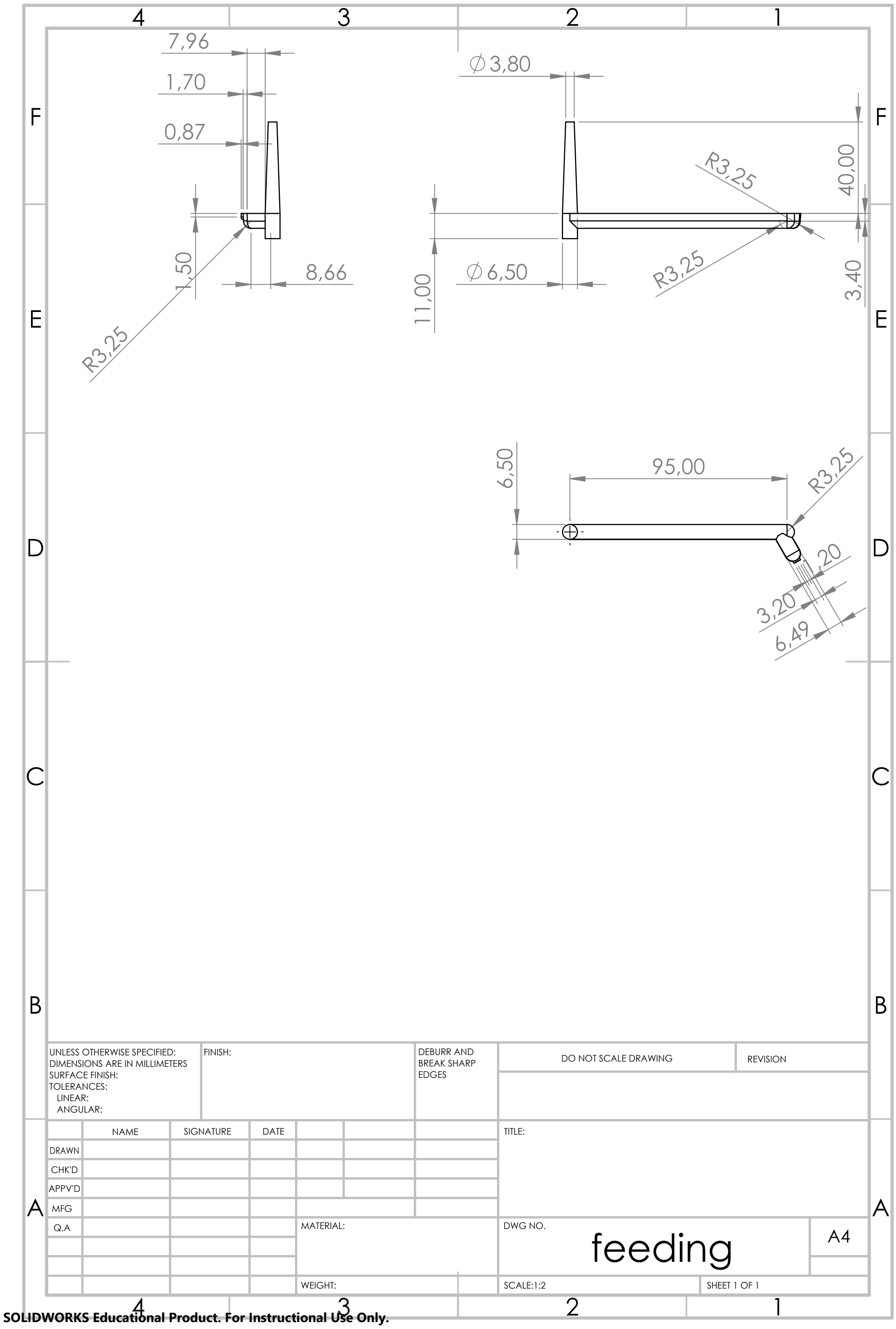

Figure 45: 2D Drawing of the feeding system.

Appendix I. Apparent order Moldex3D

As stated in Section 6 the Richardson's extrapolation method applied to Moldex3D® was made concerning only the two more refined meshes, because the results obtained with the commercial software did not present convergence of errors, and in some cases, did not present monotonic behaviour as well. In this Appendix, we will demonstrate the results obtained for pressure field using the usual Richardson's extrapolation (Equation (84)), which means calculating the apparent order (Equation (83)). Therefore, Table 30 presents the values of the pressure field for all levels of refinement, the extrapolated value (RE), the relative errors for each mesh, and the apparent order obtained at the switch-over point.

Table 30: Maximum pressure values obtained with the commercial software Moldex3D® and the open-source software OpenFOAM® for the switch-over point. The Richardson's extrapolated (RE) values, the relative errors and the calculated apparent order are also presented.

Mesh	Moldex3D®		
	Maximum (MPa)	*Error (%)*	*Apparent order*
M1	7.18	33.5	0.54
M2	8.32	54.8	
M3	9.66	79.7	
M4	11.61	115.9	
RE	5.38	—	—

Table 30 shows that although the values of pressure present a monotonic behaviour, the errors increased with mesh refinement, which is proved by the value of the apparent order. This behaviour happened for all field variables, therefore, the Richardson's extrapolation method with only the two more refined meshes was considered.