Universidade do Minho
Escola de Engenharia

Nelson Daniel Ferreira Gonçalves

COMPUTER AIDED DESIGN OF EXTRUSION FORMING TOOLS FOR COMPLEX GEOMETRY PROFILES

january 2013

COMPUTER AIDED DESIGN OF EXTRUSION
FORMING TOOLS FOR COMPLEX GEOMETRY PROFILES

Nelson Daniel Ferreira Gonçalves

UMinho | 2013

Universidade do Minho
Escola de Engenharia

Nelson Daniel Ferreira Gonçalves

COMPUTER AIDED DESIGN OF EXTRUSION FORMING TOOLS FOR COMPLEX GEOMETRY PROFILES

Doctoral Dissertation for PhD degree in
Science and Polymer Engineering and Composites

Thesis Supervisors:
João Miguel de Amorim Novais da Costa Nóbrega
Olga Machado de Sousa Carneiro

january 2013

Autor: Nelson Daniel Ferreira Gonçalves **CC:** 11913428

Email: nelson.goncalves@dep.uminho.pt / nelsondfg@hotmail.com

Título da tese

Computer Aided Design of Extrusion Forming Tools for Complex Geometry Profiles

Orientadores

João Miguel de Amorim Novais da Costa Nóbrega

Olga Machado de Sousa Carneiro

Ano de conclusão

2013

Doutoramento em Ciência e Engenharia de Polímeros e Compósitos

Nelson Daniel Ferreira Gonçalves

Universidade do Minho, 28 de janeiro de 2013

I would like to dedicate this thesis to my family.

ACKNOWLEDGMENTS

I would like to acknowledge the Portuguese Foundation for Science and Technology (FCT) for their financial support throw the scholarship SFRH/BD/43632/2008.

I express my gratitude to my advisers Professor Miguel Nóbrega and Professor Olga Carneiro for their knowledge and support. I thank Professor Miguel Nóbrega for his patient and time spent since my master's thesis. I thank Professor Olga Carneiro for her clear and simple explanations about complex phenomena.

I sincerely express my gratitude to my friend Luis Ferrás for his mathematical translations of engineering phenomena.

I thank all my colleagues in Polymer Engineering Department of University of Minho, Célio Pinto, Pedro Marques, Paulo Teixeira, Rui Novais, Gabriela Azevedo and all other persons I had the privilege of work with.

I thank my wife, Ângela, by her understanding while I work, by her support when things do not go well and by the love always shown.

ABSTRACT

In the profile extrusion, the experience of the die designer is crucial for obtaining good results. In industry, it is quite usual the need of several experimental trials for a specific extrusion die before a balanced flow distribution is obtained. This experimental based trial-and-error procedure is time and money consuming, but, it works, and most of the profile extrusion companies rely on such method. However, the competition is forcing the industry to look for more effective procedures and the design of profile extrusion dies is not an exception.

For this purpose, computer aided design seems to be a good route. Nowadays, the available computational rheology numerical codes allow the simulation of complex fluid flows. This permits the die designer to evaluate and to optimize the flow channel, without the need to have a physical die and to perform real extrusion trials.

In this work, a finite volume based numerical code was developed, for the simulation of non-Newtonian (inelastic) fluid and non-isothermal flows using unstructured meshes. The developed code is able to model the forming and cooling stages of profile extrusion, and can be used to aid the design of forming tools used in the production of complex profiles. For the code verification three benchmark problems were tested: flow between parallel plates, flow around a cylinder, and the lid driven cavity flow. The code was employed to design two extrusion dies to produce complex cross section profiles: a medical catheter die and a wood plastic composite profile for decking applications. The last was experimentally validated. Simple extrusion dies used to produced L and T shaped profiles were studied in detail, allowing a better understanding of the effect of the main geometry parameters on the flow distribution.

To model the cooling stage a new implicit formulation was devised, which allowed the achievement of better convergence rates and thus the reduction of the computation times.

Having in mind the solution of large dimension problems, the code was parallelized using graphics processing units (GPUs). Speedups of ten times could be obtained, drastically decreasing the time required to obtain results.

KEYWORDS

Profile extrusion, extrusion dies, calibrator, flow balance, cooling, computer aided design, complex geometry profiles, Finite Volume Method, unstructured meshes, code parallelisation, graphics processing units.

RESUMO

No processo de extrusão de perfis, a experiência do projetista é crucial para a obtenção de bons resultados. Na indústria, é bastante usual serem necessárias várias modificações na geometria do canal de fluxo da cabeça de extrusão para a obtenção de distribuição de fluxo equilibrada. Geralmente, o procedimento é baseado num processo de tentativa-e-erro, que consome muito tempo e dinheiro, mas funciona, e muitas das empresas de extrusão utilizam esta metodologia. A concorrência tem forçando a indústria a procurar procedimentos mais eficientes, e o projeto de cabeças de extrusão não é exceção. Para este efeito, o projeto assistido por computador parece ser uma boa via. Atualmente, os códigos numéricos de reologia computacional permitem simular escoamentos de fluidos complexos. Este facto permite ao projetista avaliar e otimizar o canal de fluxo, sem a necessidade de ter, fisicamente, uma cabeça de extrusão e fazer testes experimentais.

Neste trabalho foi desenvolvido um código numérico baseado no método dos volumes finitos, para a simulação de escoamentos de fluídos não-Newtonianos (inelásticos) e não isotérmicos, usando malhas não estruturadas. O código desenvolvido é capaz de modelar as etapas de extrusão e arrefecimento do perfil extrudido, e pode ser usado para auxiliar o projeto de ferramentas de extrusão empregues na produção de perfis complexos. Para verificar o código foram testados três problemas de referência: fluxo entre placas paralelas, fluxo à volta de cilindro e escoamento em cavidade bidimensional. O código foi usado no projeto de duas cabeças de extrusão para a produção de perfis de secção transversal com geometria complexa: um cateter para aplicações médicas e um perfil em madeira compósita para aplicações em pisos de áreas externas. A última foi validada experimentalmente. O desempenho de cabeças de extrusão para a produção de perfis simples, em forma de L e T, foi estudado em detalhe, permitindo compreender o efeito dos principais parâmetros geométricos na distribuição do fluxo.

Para modelar o arrefecimento foi desenvolvida uma nova implementação implícita, que permitiu obter melhores taxas de convergência e consequentes reduções de tempos de cálculo.

Tendo em mente a resolução de problemas de grandes dimensões, o código foi paralelizado em placas gráficas (GPUs). Com esta melhoria obtiveram-se acelerações de cálculo em cerca de dez vezes, diminuindo drasticamente os tempos necessários para efetuar as simulações.

KEYWORDS

Extrusão de perfis, cabeças de extrusão, calibrador, equilíbrio de escoamento, arrefecimento, projeto assistido por computador, perfis com geometrias complexas, Método dos Volumes Finitos, malhas não-estruturadas, paralelização de código, unidades de processamento gráfico.

PUBLICATIONS

Contents of this document appeared or will appear in the following publications:

- Computer Aided Design of Extrusion Dies for Complex Geometry Profiles; Nelson D. Gonçalves, Ângela M. Ribau, Olga S. Carneiro, João M. Nóbrega; Materials Science Forum; Vols. 730–732; pp 495-500; 2013.
- Modeling the Calibration/Cooling Stage of Complex Thermoplastic Profiles; Ângela M. Ribau, Nelson D. Gonçalves, Carolina Silva, João M. Nóbrega; Olga S. Carneiro; Materials Science Forum; Vols. 730– 732; pp 537-542; 2013.
- Design of complex profile extrusion dies through numerical modeling; N. D. Gonçalves, O. S. Carneiro, J. M. Nóbrega; Journal of Non-Newtonian Fluid Mechanics; Vol 200; pp 103-110; 2013.
- Design and optimization of an extrusion die for the production of wood-plastic composite profiles; N. D. Gonçalves, P. Teixeira, L. L. Ferrás A. M. Afonso, J. M. Nóbrega and O. S. Carneiro; submitted to Polymer Engineering and Science.
- Using the GPU to design complex profile extrusion dies; N. D. Gonçalves, S. P. Pereira, L. L. Ferrás, J. M. Nóbrega, O. S. Carneiro; submitted to Computer Methods in Applied Mechanics and Engineering.
- Design of profile extrusion dies: the role of pre-parallel zone; N. D. Gonçalves, L. L. Ferrás, O. S. Carneiro and J. M. Nóbrega; in preparation for submission in Journal of Material Processing Technology.
- Cooling stage in profile extrusion: an implicit numerical approach; N. D. Gonçalves, L. L. Ferrás, O. S. Carneiro and J. M. Nóbrega; in preparation for submission in International Journal of Heat and Mass Transfer.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

BC Boundary Condition

CDS Central Differencing Scheme

CFD Computational Fluid Dynamics

CV Control Volume

ES Elemental Section

FDM Finite Difference Method

FEM Finite Element Method

FVM Finite Volume Method

GPU Graphics Processing Unit

TVD Total Variation Diminishing

UDS Upwind Differencing Scheme

1 INTRODUCTION

1.1 EXTRUSION

Extrusion is the process used to manufacture products in the form of continuous lengths with an constant cross section. The most common materials used in the extrusion process are polymers, metals, ceramics, concrete and foodstuffs. In the case of polymers, the inherent design freedom promoted by the employment of thermoplastic make them attractive for extrusion. Theoretically, thermoplastic profiles can be produced with any cross section suited for a specific application. A typical extrusion line for the production of thermoplastic profiles (Fig. 1.1) generally comprises an extruder, a die, a calibration/cooling system (which can have several units), a haul-off unit and a saw (or, alternatively, a coiling device).

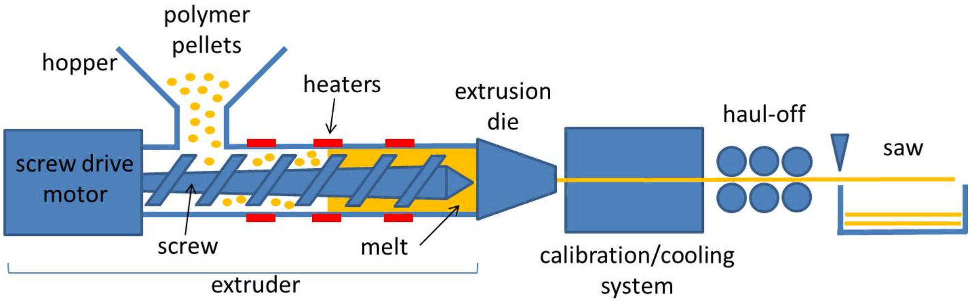


Figure 1.1: Schematic view of the extrusion process

In the polymer extrusion process, the polymer pellets are fed in to a hopper that, by gravity, feeds the barrel of the extruder. Additives can be added, mixed with the polymer pellets. Inside the extruder, the pellets are transported by one or two rotating screws. As the polymer is forced forward, it is gradually heated up to the desired melt temperature. At the extruder exit, or discharge zone, the molten material is forced through an extrusion die that shapes the melt into a specified cross-section. After the die exit, the polymer profile must be cooled, and this can be achieved by pulling it through a water bath or a calibrating/cooling system. The cooled product is then cut or coiled up onto a spool. In addition to these components, the extrusion line can have other elements, e. g., more extruders (co-extrusion), when the profile comprises more than one material, monitoring systems, ovens for thermal stabilization, etc..

1.2 EXTRUSION DIE

The main function of the extrusion die is to convert the circular flow from the extruder outlet into the required cross section.

The main objectives of the extrusion die design process, are to obtain a die that allows the production of the required profile at the highest rate while guaranteeing an adequate level of quality. For this, the minimum level of internal stresses, avoidance of rheological defects (e. g., shark skin and melt fracture) and prevention of thermal degradation of the melt have to be guaranteed (Carneiro et al., 2001). In order to achieve these objectives, the design of these extrusion tools must consider (Carneiro et al., 2001):

- the maximum flow rate above which shark skin occurs;
- the maximum angle of convergence above which melt fracture in extension-dominated flows occurs;
- the correction of the cross-section of the parallel zone to anticipate post-extrusion effects (stretching, swelling and shrinkage);
- the design of the flow channel that assures a balanced flow distribution at the die exit, i. e., to attain the similar average melt velocities in the various zones of the die exit cross-section, in order to allow the production of the required geometry;
- the control over both the total pressure drop and the appearance of hot spots (local increases in melt temperature) resulting from excessive viscous dissipation.

The design of the extrusion die flow channel geometry needed to produce the desired profile, is a very complex task that demands advanced knowledge about material rheology. Knowledge about heat transfer phenomena, as well as extensive experience in extrusion processing and related operating conditions adopted, are also needed for a successful design of extrusion dies (Kostic and Reifschneider, 2006; Carneiro et al., 2001).

The complex behavior of the polymer melts, due its viscoelastic nature, and the large number of phenomena involved in the polymer extrusion process, make the design task to be strongly dependent on the designer's knowledge and experience. This limitation often requires several trials until the desired geometry is achieved (Carneiro et al., 2001) resulting in processes with high time, material and equipment consumption (Mu et al., 2010; Nóbrega, Carneiro, Pinho and Oliveira, 2003).

In spite of significant advances in the computation capabilities and developments on modelling of complex flow and heat transfer processes, the extrusion die design is still more an art than a science (Kostic and Reifschneider, 2006).

Using commercially available computational tools (*ANSYS Polyflow*, *Dieflow*, *HyperXtrude*, *FLOW 2000*, *PROFILECAD*) to simulate the melt flow throw the extrusion die, enables to achieve considerable gains, since they allow the progressively substitution of the experimental trial-and-error procedures by computational based ones. However, these commercial software leave to the designer the decisions needed to successively improve the flow channel's geometry until acceptable results are achieved. The commercial codes, usually do not include relevant phenomena, e. g., die swell. Additionally, such numerical codes are not user-friendly and require large computational times, which do not fit the industrial needs (Carneiro et al., 2001).

As mentioned before, the optimization of these tools starts with a initial (trial) design which is corrected iteratively until a satisfactory flow balance is achieved. These corrections can be done based on designer's experience. This iterative process can not be automatized, since it depends on the designer's knowledge and intervention.

In order to make this process independent of the persons involved, computational modelling codes can be coupled with optimization procedures. For that purpose, the evaluation of each trial die performance should be done by an objective function, whose value should quantify the tool performance (Ettinger et al., 2004). Until now, several objective functions have been used, according to the different applications, aiming the optimization of several processes and product properties as described in (Szarvasy et al., 2000). As an example, we have the flow balancing, studied in Koziey et al. (1996) or the studies made by Nóbrega and Carneiro (Nóbrega, Carneiro, Pinho and Oliveira, 2003; Carneiro, Nóbrega, Oliveira and Pinho, 2003; Nóbrega, Carneiro, Pinho and Oliveira, 2004), where the objective function gives an indication of the homogeneity of the flow distribution.

Optimization algorithms, such as gradient-based or evolutionary methods, can be employed in this class of optimization processes. In simple profiles the number of design variables is usually between 5 and 10, and for that it is not necessary to develop special optimization algorithms, unlike the real industrial applications, where the number of design variables can be circa 50 (Ettinger et al., 2004). The optimization strategy can take advantage of the particular problem at hand and can usually improve the performance of the extrusion die, as well as its robustness. The simplex method, which Nóbrega et al. (Nóbrega, Carneiro, Pinho and Oliveira, 2003) applied in their work dealing with simple three dimensional die geometries, seems to be a better strategy. When dealing with complex geometries, such as, for example, window frame profile dies, the need for an effective optimization strategy is crucial for any automated die design procedure (Ettinger et al., 2004).

Mu et al. (2010) presented an optimization strategy for die design process, based on finite element simulation, the back-propagation neural network and the non-dominated sorting genetic algorithm II (NSGA-II). The strategy's performance was used to evaluate the design of an annular extrusion die, where the objective was to distribute uniformly the outlet flow and to reduce the extrudate swell of the extrudate.

For the processing design of steady viscous non-Newtonian fluid, Smith (2003) proposed an optimization model based on numerical simulation and sensitivity analysis.

For the design of extrusion dies to the production of profiles with relatively complex geometries that can be divided into simple geometries, such as rectangles, Carneiro et al. (2001) presented an optimization approach based on the inclusion of a pre-parallel zone between the transition zone and the parallel zone. In this case, the flow balance was achieved by the optimization of the pre-parallel zone length, which could be automatically and easily changed.

To the design of the same type of extrusion die geometry, Michaeli et al. (2001) introduced a new optimization methodology, combining finite element methods with net theory, which accelerated the optimization process. This method can help determining where and how to adjust the die flow channel in order to achieve a uniform distribution of the polymer melt flow.

The current available studies are related to profiles with relatively simple geometries or 2D modelling. In order to perform the optimization of dies required in industry, it is necessary to expand the current studies to more complex geometries.

1.3 calibrator

The viscoelastic behavior of polymer melts and the unavoidable fluctuations of the operating conditions (which affect the polymer melt rheological behavior and flow dynamics), makes very difficult the procedure of producing an extrudate with the exact required cross section (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004; Nóbrega, Carneiro, Pinho, Oliveira and Covas, 2003). Furthermore, as the profile advances along the extrusion line, a variety of external forces, such as friction, buoyancy, gravity and compression act on it, which can cause important deformations, unless efficient cooling ensures enough profile strength (Kleindienst, 1973; Michaeli, 2003). In order to ensure that the incoming melt is properly shaped, the calibration/cooling step has a double objective (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004): prescribing the final dimensions of the profile and promote its fast cooling in order to solidify the outer layers, ensuring sufficient rigidity during the remainder production steps (Kleindienst, 1973).

The average profile temperature at the calibrator outlet must be lower than the solidification temperature of the polymer in order to avoid subsequent remelting (Kleindienst, 1973). Also the temperature gradient of the profile cross sections should be minimized (Nóbrega, Carneiro, Pinho, Oliveira and Covas, 2003), so that adequate morphology development is promoted and residual thermal stresses levels are minimized (Fradette et al., 1995; Brown and Enterprises, 2000). Hence, the main objectives of the calibration stage are the fast and homogeneous cooling of the profile (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004).

The calibration can be done applying internal pressure or external vacuum. It can also be wet and/or dry, i.e., with direct contact with cooling medium (generally, water) and the profile or indirect through the calibrator material (Endrass, 1993; Schiedrum, 1983). Several calibrators can be used in series, separated by air zones (Kurz, 1988; Pittman et al., 1994), where the temperature tends to homogenize, leading to the minimization of internal thermal induced stresses and increasing the heat transfer efficiency of the next calibrator. The vacuum-assisted dry calibration has proved to be particularly reliable for high-speed profile extrusion (Endrass, 1993). A combined wet/dry system, with a dry vacuum block unit followed by several water-cooling tanks can also be used (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004).

The parameters that influence the thermal performance of the calibration system (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004) may be grouped as follows:

- system geometry: number of calibrating units, unit length, separating distance and layout of the cooling channels (the latter involves quantities such as, number, diameter, type of arrangement, distance between consecutive channels and distance to the profile) (Schiedrum, 1983; Sheehy et al., 1994) and orientation relative to the extrusion direction (Nóbrega, Carneiro, Pinho, Oliveira and Covas, 2003);

- cooling conditions: temperature of the inlet water, flow rate, flow direction and wet versus dry contact with the profile (Kleindienst, 1973; Michaeli, 2003);

- vacuum conditions: number and location of vacuum holes and vacuum pressure;
- extrusion conditions: mass flow rate and cross temperature profile field at die exit;
- polymer thermo-physical properties: thermal diffusivity and thermal expansion coefficient;
- properties of the calibrator material: thermal conductivity and surface roughness;
- profile cross section: thicknesses, number and location of hollow sections, etc..

The complex geometries of profiles required by real problems, together with the above mentioned parameters, demand the use of numerical methods to determine the time evolution of the system temperature distribution and allow its improvement (Nóbrega, Carneiro, Pinho, Oliveira and Covas, 2003).

When modelling the heat exchange on a calibrator, an additional difficulty is the selection of the proper heat transfer coefficient, h, that accounts for the heat transfer between the plastic profile and the calibrator. Fradette and Tanguy (1996) have shown, experimentally, that h might vary between 10 and $1000 W m^{-2} K^{-1}$, depending on the location along the calibrator. Placek et al. (2000) proposed different h values, depending on the nature of the polymer contact with the cooling medium (calibrator or water).

Despite the practical relevance of calibrating and cooling systems, their study has attracted relatively little attention in the scientific literature. The majority of the available studies are related with the calculation of the time evolution of the extrudate temperature (Pittman et al. (1994), Placek et al. (2000)). The work of Fradette et al. (1995), uses a previously developed model (Sheehy et al., 1994) that is integrated in an optimization routine and used to determine the optimal location and size of the cooling channels.

Nóbrega, Carneiro, Covas, Pinho and Oliveira (2004) present and validate a 3D code based on FVM that models the thermal interchanges between the polymer and calibrator over the course of the calibration and cooling stage. In the same work, a study of the influence of the boundary conditions, geometrical and operating parameters on the performance of the cooling stage is presented. The existing works are limited to simple geometries, being needed to extend them to more complex geometries.

1.4 MODELLING

The need to study some phenomena, such as the weather, in order to be able to forecast it lead to the creation of abstract concepts to represent a system and the factors that influence it. This creation process is known as numerical modelling. In the numerical modelling process mathematical equations connecting several parameters can be used, but usually, simplifications of the reality have to be made (Maria, 1997). These approaches can be made to study a particular part or feature of the world and often allow to solve complex problems, which resolution is unknown or time consuming. Based on models, numerical codes can be developed in order to simulate and study phenomena, in order to better predict the reality.

In the case of fluids, the analysis of systems involving fluid flow, heat transfer and associated phenomena (such as chemical reactions), performed by computer-based simulation, comprise the Computational Fluid Dynamics (CFD) field. Some of the most relevant applications of CFD (Versteeg and Malalasekera, 2007) are: aerodynamics and hydrodynamics of vehicles, chemical processes, combustion, electrical and electronic engineering, distribution of pollutants, weather forecast and biomedical engineering.

There are several reasons to use CFD instead of laboratory experiments. The most important is perhaps the economic reason, because there is no waste of physical resources in numerical simulations. Additionally, reduced time used in the numerical simulations, when compared with laboratory experiments, and the consequent possibility of testing more conditions in the same time interval, make the use of CFD an attractive and very powerful tool.

In addition to the reasons referred before, the use of CFD allows the achievement of data that could not be obtained during experimental work. As examples, one can mention the study of the reactions that happen inside a nuclear reactor (where it is impossible to perform measurements in place) and the study of systems under hazardous conditions (Versteeg and Malalasekera, 2007). The CFD field allows also the study of systems where controlled experiments are difficult or impossible to perform as, for example, the flow around an aircraft during flight: when it is performed in a laboratory, the aircraft is next to the ground, which is unrealistic and affects the flow distribution around it; on the other hand, with the aircraft flying it is not possible to collect the same data. In polymer processing, it is also very difficult/impossible to measure, for example, the temperature distribution inside the flow channel without affecting the flow dynamics.

An additional advantage of the use of CFD is the almost unlimited capacity on the detail level of the results, since the data can be taken everywhere in the domain, safely and without affecting the results, unlike the measurements made in the laboratory. The ability to refine the mesh on zones with more interest and the possibility to get values in very large systems, are other advantages of the use of CFD numerical codes.

When complex non-Newtonian fluids are treated, e. g. molten plastics and polymer solutions, and the simulations of its flow through complex geometries are needed, the process become more challenging (Walters and Webster, 2003). In this case, the Navier-Stokes equations to Newtonian fluids are not enough to model the behavior of non-Newtonian fluids, and other models have to be considered in order to predict the rheometrical behavior of this type of more complex flows.

The commercial CFD codes devoted to polymer processing, *ANSYS Polyflow*, *Dieflow*, *FLOW 2000* and *PROFILECAD*, are based on the Finite Element Methods (FEM). This method allows to perform computations in complex geometries, as opposed to Finite Difference Method (FDM). An alternative method, the Finite Volume Method (FVM) (Patankar, 1980; Ferziger and Peric, 2002), starts from the integral form of the conservation equation. The domain is divided into a finite number of contiguous control volumes (CV) and the governing equations are discretized for each CV. The values of the variables are saved in the centroid (geometric center) of each CV, being their values on the CV faces obtained by interpolation. The volume and surface integrals are approached using algebraic formulas, leading to one equation per CV. The FVM can be applied to any kind of mesh, making it suitable for problems with complex geometries. The method is conservative, since the surface integrals (that represent advective and diffusive fluxes) are the same on CV shared surfaces. The FVM is easier to understand because of the physical meaning of each term that comprises the discretized equations, thus is usually referred as the approach preferred by engineers.

1.5 MESHES

The mesh is a partition of the physical domain into smaller elements (cells or control volumes) that allows to transform a continuous mathematical problem into a discrete one, with a variable (or set of variables) stored in each CV (Gonçalves, 2007).

1.5.1 *Orthogonal meshes*

The simplest meshes are orthogonal (or Cartesian), where the lines that separate the elements are parallel to the co-ordinate system axis.

Since many engineering problems involve complex geometries, the orthogonal meshes are not a good option to fit such geometries, for example, the geometry for the flow involving irregular geometries, is not well discretized using an orthogonal mesh as can be seen in Fig. 1.2. In this case, to calculate the flow around several cylinders, the mesh must be superimposed with geometry of each cylinder, leading to several disadvantages due to the inaccuracy of the boundary description. The cells inside the cylinder domain become unused (in this case, they are approximately 25% of the total number of elements). Another drawback of this type of meshes is the fact that a local refinement implies a global refinement (Versteeg and Malalasekera, 2007).

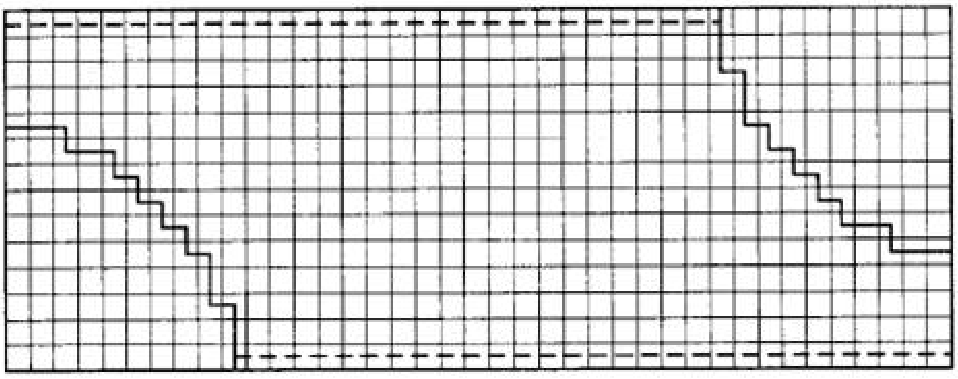


Figure 1.2: Orthogonal mesh used in the discretization of the geometry around two cylinders (adapted from Versteeg and Malalasekera, 2007).

As for the Cartesian coordinate system, the polar, cylindrical and spherical coordinate systems have similar limitations in irregular geometries.

1.5.2 *Non-orthogonal structured meshes*

In order to avoid the limitations of the orthogonal meshes, non-orthogonal ones can be used instead. Structured non-orthogonal meshes can fit (*body-fitted grid*) the geometry boundaries with curvilinear coordinates (Fig. 1.3). In this case, all the elements of the mesh belong to the domain (mesh and domain are superimposed) and, therefore, the computer resources are used more efficiently.

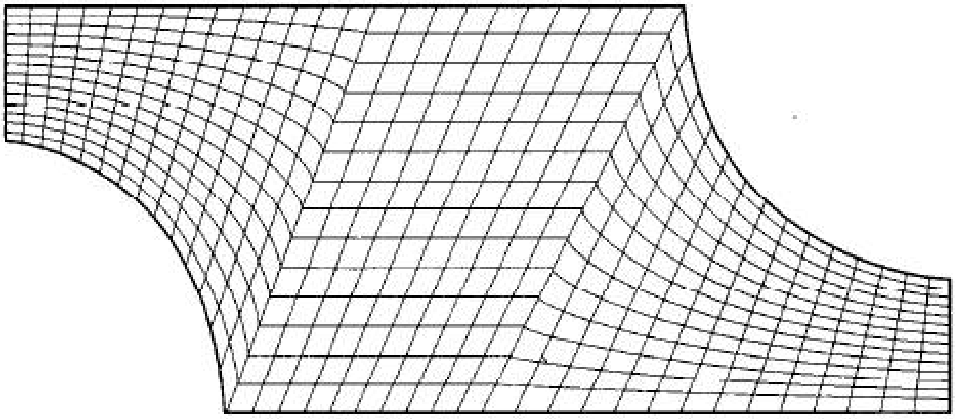


Figure 1.3: Structured mesh body-fitted mesh used in the discretization of the geometry around two cylinders (adapted from Versteeg and Malalasekera, 2007).

The structured meshes can be of H-type, O-type or C-type (Blazek, 2001; Cebeci et al., 2005). The designation of the mesh type derive from the geometry of the mesh lines. The H-type meshes (Fig. 1.4a), when projected on a rectangle, have different logical boundaries: east, west, north and south. In O-type meshes (Fig. 1.4b), a set of lines were "unlimited", i. e. if these lines are considered coordinate lines, the coordinates would increase indefinitely. To prevent this from happening, an artificial "cut" is made in the mesh, to set the coordinates origin and limit. In C-type meshes (Fig. 1.4c) vortices on different portions of the mesh are coincident, leading to the introduction of an artificial cut, similar to the O-type meshes case. This type of meshes are commonly used with sharp geometries, allowing a good accuracy.

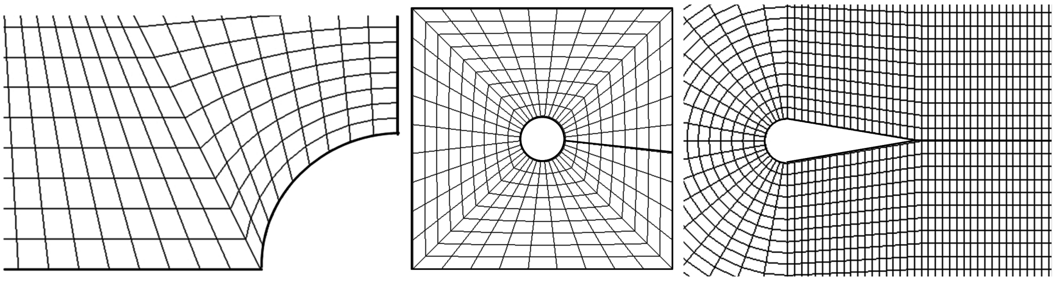

(a) Mesh type H. (b) Mesh type O. (c) Mesh type C.

Figure 1.4: Structured mesh types

The flexibility given by non-orthogonal structured meshes is very useful because:

- all geometry details can be incorporated accurately;
- the mesh properties can be controlled to collect more information on regions of interest (Hoffmann and Chiang, 2000).

Despite the advantages of non-orthogonal structured meshes, some drawbacks exist:

- still difficult and time consuming to generate;
- more information has to be saved and more calculations are required (e.g. gradients evaluation);
- if the solution domain cannot be readily mapped into a rectangle (in 2D) or a rectangular parallelepiped (in 3D) this can result in skewed grid lines causing unnecessary local variations (Versteeg and Malalasekera, 2007);

- unnecessary grid resolutions can result in cases where mapping is difficult;
- mapping is sometime impossible for complex 3D geometries with internal objects/parts.

1.5.3 *Block-structured meshes*

In order to bypass some of the limitations associated with the use of structured meshes, *block-structured meshes* were developed. With this type of meshes, the geometry is divided into several parts (blocks) and each block can be discretized with different types of structured meshes (Fig. 1.5). The mesh generated in each block can have different standard size of elements, allowing mesh refinement in specific areas, without the need for mesh refinement in zones were the gradients are small.

In faces shared by two blocks, when the elements in each block have different sizes (Fig. 1.6), a special treatment is required (Ferziger and Peric, 2002), since the interface must be treated in a totally conservative way. In Fig. 1.5 is shown a mesh that comprises three blocks, with different mesh refinements.

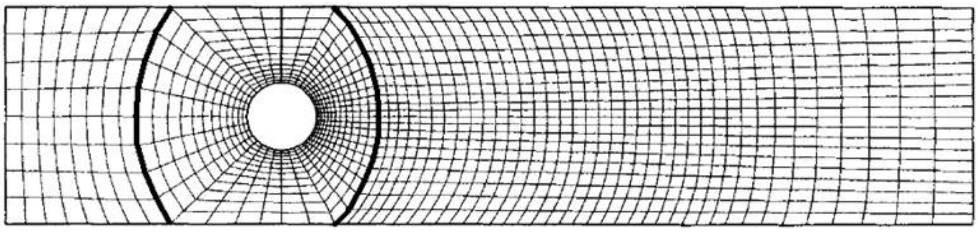


Figure 1.5: Block-structured mesh with three blocks (adapted from Ferziger and Peric (2002))

In Fig. 1.6, a mesh that comprises five blocks is used to fit the geometry of an airfoil. In the blocks farthest from the airfoil, the mesh elements are larger reducing the required computational times. In the three middle blocks, the elements are smaller because higher gradients are expected around the airfoil. In the block around the airfoil, the elements have different distribution in order to obtain a better fitting.

The numerical codes that use block-structured meshes are more complex because of the special treatment required for the block interface. However, standard procedures

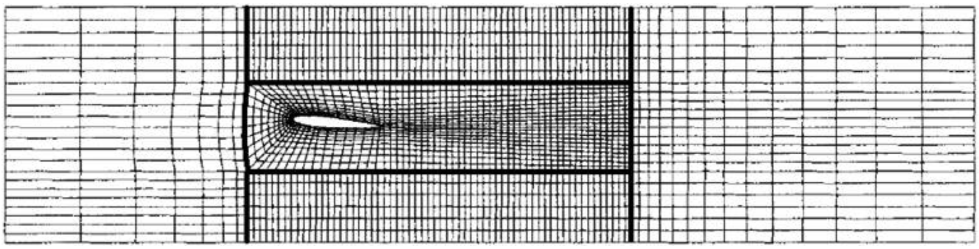


Figure 1.6: Block-structured mesh with five blocks (adapted from Ferziger and Peric (2002))

can be applied inside each block. In some codes the solvers are applied in a block-wise manner (block by block with overall final iterations to unify boundary conditions).

1.5.4 *Chimera meshes*

The chimera meshes (or composite meshes), are composed by overlapping blocks. Each mesh can have different refinements (Ferziger and Peric, 2002). This method is used, for example, when a body moves through a domain, with one mesh following the moving body while the remaining part of the domain stays static.

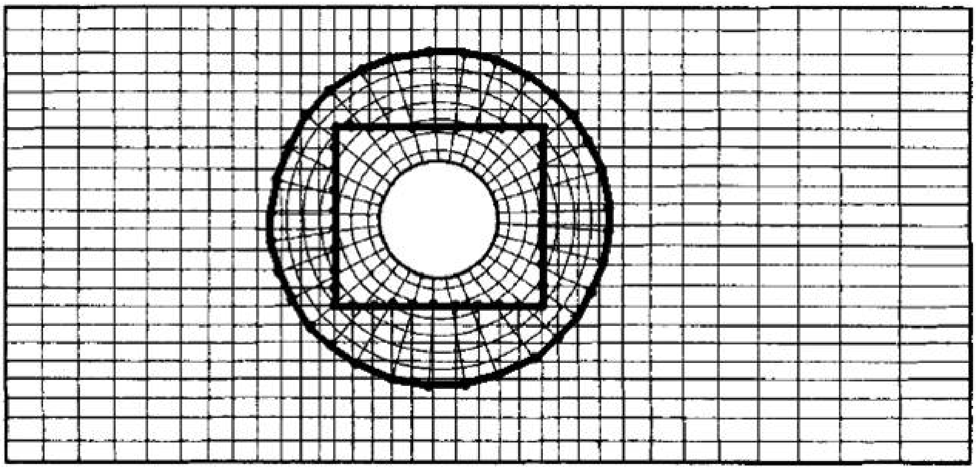


Figure 1.7: Chimera meshes (adapted from Ferziger and Peric (2002))

In the overlapping region, the boundary conditions between blocks are obtained by interpolation. The disadvantage of this type of meshes is that the conservation is not guaranteed at the boundaries between blocks.

1.5.5 *Unstructured meshes*

The unstructured meshes are more flexible, since they can adapt to geometry boundaries with no restrictions. This type of meshes can be used with any discretization method but adapt better to the FVM. The cells (or control volumes) can have different shapes and there are no restrictions in the number of neighbors. The most used element shapes are triangles and quadrilaterals, in 2D geometries, and tetrahedrons, prisms or hexahedrons, in 3D ones. In the unstructured mesh arrangement we are not restricted to one particular element type, being possible to use a combination of cell shapes.

Notice that for geometries with a parallel zone (Nóbrega, Carneiro, Pinho and Oliveira, 2003) (Fig. 1.8) the use of triangular prisms can spend circa 20% of the computational resources when compared to the use of tetrahedrons.

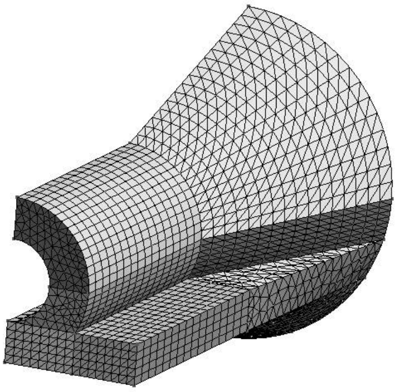


Figure 1.8: Unstructured mesh constituted by tetrahedrons and triangular prisms.

The disadvantages of unstructured meshes result from their lack of structure. The irregularity of the associated data structures (Thompson et al., 1999) requires more information about geometry elements and irregular computational cells, leading to higher computational times. In structured meshes there is no need to specify the index of the neighbors, because they can be obtained with algebraic operations on indices; in unstructured meshes the neighbors indices must be specified for each face. The system of equations obtained with unstructured meshes has not a banded matrix (Versteeg and Malalasekera, 2007). Different number of neighbors for each element, resulting from geometrical constraints, result in to a more complex structure of the system of equations, requiring the development of adequate solution procedures.

1.6 code parallelization

In recent years, computational fluid dynamics (CFD) has been gaining acceptance in industry as a design tool. Advances in algorithms development and computational hardware, have led to more complex modelling of fluid flows (Alonso et al., 1994).

Increasingly complex fluid flow models require high performance computing facilities. A cost effective solution for problems of this type requires fast CPUs, large internal memory and the use of parallel computing (Alonso et al., 1994; Tezduyar et al., 1993; Topping and Ivanyi, 2010; Topping, 2002, 2000; Mattson et al., 2005; Stone, 1992; Lazou, 1988; Dongarra et al., 2003; Fox et al., 1994).

A possible strategy to parallelize a numerical code is by domain decomposition into sub-domains. (Kruis, 2007, 2004; Magoules, 2010, 2008; Glowinski, 1997; Kruis et al., 2002). In these, elements of each sub-domain are treated independently of the other sub-domains and communication between sub-domains is performed through cells surrounding each sub-domain boundary. Reuther et al. (1996) uses an adjoint formulation implemented in parallel on distributed memory computers, with a domain decomposition approach and the Message Passing Interface (MPI) Standard to optimize the aerodynamic shape of a supersonic aircraft. Dellagiacoma et al. (1995) parallelized a previous code in a multi-block environment, thus multiple copies of the code can work concurrently on different grid blocks. Farhat et al. (1994) analyzes the performance of Finite Element Tearing and Interconnecting (FETI) method, a domain decomposition algorithm for the solution of self-adjoint elliptic partial differential equations and later (Farhat et al., 1998) extend the work to shell problems and analyzed the performance results of the parallel implementation. Bhardwaj et al. (2000) analyzed the numerical and parallel scalability of FETI method. Dostál et al. (Dostál, Gomes Neto and Santos, 2000; Dostál, Neto and Santos, 2000; Dostál, 2009; Dostál et al., 2007) presented several results concerning the development of optimal algorithms for the solution of large bound and/or equality constrained quadratic programing (QP) problems with high parallel scalability.

The parallelization can also be done employing multi-grid techniques (Wördenweber, 1984; Frey and George, 2010; Topping and Khan, 1996; Magoules, 2008; Walshaw and Cross, 2008) that enhance the convergence rate of a scheme by performing computations on a series of increasingly coarser meshes (Chesshire and Jameson, 1989; Versteeg and Malalasekera, 2007), and implicit residual smoothing (Alonso et al., 1994). Message passing becomes necessary in this new environment and severely limits the performance of processes that are inherently communication intensive (Alonso et al., 1994).

A multi-computer architecture, is a possibility that can be used for parallel computations (Sterling et al., 1999). It is a system that usually consists of one server node, and one or more client nodes connected a computer network. It is a system built using commodity hardware components, like any PC capable of running a Unix-like operating system, with standard Ethernet adapters, and switches. It does not contain any custom hardware components and is trivially reproducible. The Open Multi-Processing (OpenMP) (*OpenMP*, 2013) is an Application Programming Interface (API) that supports multi-threaded execution of a program for shared memory architectures (Jost et al., 2003). OpenMP provides a fork-and-join execution model with a thread that starts the program execution and creates a team of other threads of which is the master thread. All threads execute the parallel region statements, after what a synchronization is made. An existing code can be easily parallelized, placing OpenMP directives around time consuming loops with no data dependencies, leaving the remaining code unchanged. The OpenMP drawbacks are the workflow optimization and memory access that are not easy to do. The Message Passing Interface (MPI) (*MPI*, 2013) is a standardized and portable message-passing system that allows communication between processes, by calling library routines that send and receive messages (Jost et al., 2003). The MPI advantage is the complete control over data distribution and process synchronization. The MPI drawback is the need of fair amount of restructuring of sequential applications.

Recently, the GPUs, that originally were developed to process images on the computer games industry, started to be used for other scientific computing, since they have a large number of multiprocessors. The GPUs availability, low cost and its capability to compute data in parallel, make them an attractive alternative to solve problems that can be parallelized, like those found in CFD problems. Recently, some linear algebra operations, like matrix-vector multiplications, have been implemented on GPU (Krüger and Westermann, 2003). The first generation of GPUs, the G60 architecture introduced by NVIDIA in 2006 (Kirk and Wen-mei, 2010; NVIDIA, 2013), was limited to single precision calculations causing some limitations, but the second generation, GT200 (NVIDIA, 2013), allow double precision computations which is the one usually used on scientific codes. The speedups achieved with GPU, 40x for simple geometries (a single block cube geometries with freestream boundary conditions on all faces) and 20x for complex geometries (flow around an airfoil) (Elsen et al., 2008), show that this is an attractive alternative to solve CFD problems. In this frame, Pereira (Pereira, 2011) developed a numerical code able to deal with complex geometries, since it uses unstructured meshes, but limited to 2D problems. Brandvik and Pullan (2008) solved an Euler flow in 2D and 3D geometries and compared the time required to solve a problem using a CPU implementation with that required by a GPU implementation. In the 2D solver, the GPU implementation was 29 times quicker whereas in the 3D solver it was 16 times faster than the implementation on the CPU. Goodnight et al. (2005) implemented the multi-grid method on GPUs to solve a heat transfer problem for a fluid mechanics application, and obtained a speedup of 2.3 times when compared with a CPU implementation. Liu et al. (n.d.) solved a 3D fluid dynamics problem in a complex environment using GPU to have a real-time solver. Since GPU have different memories with different features, a memory management can be done to decrease the computation time. Taking this into account, Corrigan et al. (2009) improved the effective memory bandwidth by reducing the total global memory access.

1.7 MOTIVATION

The production of extruded profiles requires the definition of several parameters that influence the process, namely the geometry of the flow channel and the operating conditions adopted during extrusion (Carneiro et al., 2001). The definition of such parameters can be based on experimental trial-and-error procedures, using the designers experience, but are very resources consuming (Szarvasy et al., 2000), boosting the development of modeling softwares (*ANSYS Polyflow*, 2013; *Dieflow*, 2013; *HyperXtrude*, 2013; *FLOW 2000*, 2013; *PROFILECAD*, 2013) to aid these decisions.

In the last years the research team of the University of Minho, where this work is framed, has developed numerical codes to aid the design of tools employed in the production of polymeric thermoplastic profiles (Carneiro et al., 2001; Carneiro, Nóbrega, Pinho and Oliveira, 2003; Carneiro, Nóbrega, Oliveira and Pinho, 2003; Carneiro et al., 2004b,?; Nóbrega et al., 2000a; Nóbrega, Pinho, Oliveira and Carneiro, 2004; Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004; Nóbrega, Carneiro, Pinho and Oliveira, 2003, 2004; Nóbrega and Carneiro, 2006; Nóbrega et al., 2008). These numerical codes were used to analyze the influence of geometric parameters and processing conditions. Aiming to automatically optimize the forming tools and processing conditions, the developed codes included structured mesh generators and optimization routines that are coupled with fluid flow and heat transfer solvers. The limitations of structured meshes, due to their difficulty to fit complex geometries and its inadequacy to define local mesh refinements, motivated the development of a numerical code based on unstructured meshes.

With the increase of complexity of industry problems, arose the need to deal with more complex geometries and consequent increase in the number of elements needed to discretize such domains. To discretize geometries as the flow around a airfoil, meshes with several millions of elements are used (Elsen et al., 2008) leading to high computational times of several hours just for one parameters set. When more complex geometries need to be studied, e. g. a complete aircraft, the computational times, in addition to memory requirements, become prohibitive. Even for simple geometries, the need of several simulations required, for example, in optimization processes, can be problematic. By the aforementioned reasons, it is essential to find out alternatives to achieve the desired results faster, in addition to accuracy aspects.

As referred, one of the most relevant problems to be faced in the design of forming tools is the flow balance. Profiles constituted by complex sections with different thicknesses, and, therefore, different local flow restrictions, require a careful design of the die flow channel in order to get a balanced flow distribution. Flow modeling tools are a powerful tool to help achieving this balance through the detailed analyses of flow distribution they can provide.

Despite the unquestionable value of numerical tools, they are not available to the majority of the designers, being the trial-and-error procedures often used to design the extrusion tools. Since these processes rely essentially on the designers experience, some prompt relations between the profile thicknesses and the die geometry characteristics needed to balance the flow at outlet, would be useful.

The cooling stage, since is a process that depends on heat distribution in domains (polymer profile and calibrator), usually comprises an additional iterative process to link the heat exchange between both domains. The heat distribution is solved independently in both domains, assuming a temperature value at the contact boundary that must be iteratively updated. In order to avoid divergence problems, a relaxation factor has to be used, which can result in longer computational times.

1.8 OBJECTIVES

This work aims to enlarge the capabilities of the currently available design tools, in order to make them appropriate to deal with complex geometries, for which the ability to deal with unstructured meshes is indispensable.

The new numerical code must be able to model the flow of incompressible and generalized Newtonian fluids under non-isothermal conditions. To model the flow, the continuity and linear momentum conservation equations must be solved, using a SIMPLE based approach, and the Arrhenius law to account for the viscosity temperature dependence. The shear viscosity dependence on shear rate must be modelled by an adequate rheological model. To obtain the temperature distribution, the energy equation must also be solved.

The new code should also be able to predict the flow distribution at the die outlet, which, together with an evaluation function and an optimization strategy, will guide the achievement of a balanced flow channel design.

Since numerical simulations of problems with increasing number of elements can spend several hours to run, a strategy to reduce the computational time should be implemented. Parallelizing the new numerical code, is then a requirement and the GPU, a recent device used to speedup CFD softwares, is an appropriate solution.

For those who have not access to this type of modeling codes, it is important to generate design guidelines relating the flow channel geometry and flow distribution. For this sake, two elementar geometries, L-shaped and T-shaped dies, should be analyzed since a large number of complex geometries can be decomposed into these two simple geometries.

In term of the design of calibrators, modeling the heat exchange between the polymer profile and the calibrator is required. The usual explicit approach employed to model this process uses relaxation factors to avoid divergence problems, leading to long computational times. Therefore, methodologies aiming the decrease of the computational time should be studied.

1.9 thesis structure

In the first chapter of this thesis, the main concepts considered in this work are detailed and the motivation and objectives are exposed.

In the second chapter, the governing equations used to model the nonisothermal, non-Newtonian (inelastic) fluid flows are described and the details of the numerical procedure employed are provided.

The following five chapters are based on papers accepted or submitted for publication in relevant scientific journals.

The first paper (Chapter 3) describes the developed numerical code, verifies its implementation and presents a case study, where it is used to design a profile extrusion die for the production of a complex cross-section profile. The results obtained show that the developed numerical code is able to deal with complex geometry problems, being thus a valuable tool to aid the design of extrusion dies.

In the second paper (Chapter 4) the application of the developed numerical code on the design of an extrusion die to produce a wood-plastic composite profile is analyzed and the numerical results obtained are compared with experimental data.

The third paper (Chapter 5) presents the parallelization of the code in the GPU and the speedup obtained with that improvement. The results obtained allowed to conclude that with increasing number of cells, the parallelization version becomes faster than the serial one, allowing the achievement of a speedups of circa 7x.

The fourth paper (Chapter 6) presents a study where relations between the thicknesses ratio and the corresponding parallel zone length of extrusion dies used to produce "T" and "L" shaped profiles required to balance the flow are devised. As previously referred, the studies performed are expected to generate useful information for the designing of more complex geometries.

The last paper (Chapter 7) studies the calibration stage of profile extrusion. This phase involves the heat exchange between the plastic profile and the metal calibrator, a multi-domain problem. In traditional approaches, this process is modeled in an explicit iterative manner. In this work a novel implicit formulation is devised, which showed to reduce the numerical instability and the calculation time.

The conclusions of this work are summarized in Chapter 8, and finally, in the Chapter 9, suggestions for future work are pointed out.

2

THE COMPUTATIONAL RHEOLOGY CODE

In this chapter, the governing equations for non-isothermal and non-Newtonian (inelastic) fluid flows, are described, together with a detailed explanation of the numerical procedure adopted.

2.1 GOVERNING EQUATIONS

Computational rheology starts with a set of differential equations that model the physical phenomena being studied. The relevant equations required can vary, depending on the physical assumptions that are made. The relevant governing equations to model the flow and energy transfer in polymer melts are described below.

Mass Conservation Equation

The conservation of mass assumes that "the rate of increase of mass in a fluid element equals the net rate of flow of mass into the fluid element" (Versteeg and Malalasekera, 2007) and can be expressed for unsteady problems as

$$\frac{\partial \rho}{\partial t} + \frac{\partial \left(\rho u_i\right)}{\partial x_i} = 0. \tag{2.1}$$

where ρ is the density and u_i the i^{th} velocity component.

For incompressible flows the mass conservation equation (or continuity equation) is further simplified, with the inertial term being dropped,

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{2.2}$$

Momentum Conservation Equation

The momentum conservation equation, derived from "Newton's Second Law", states that "the rate of change of momentum of a fluid particle equals the sum of the forces on the particle" (Versteeg and Malalasekera, 2007). These forces can be divided into surface forces (pressure and viscous forces), commonly considered as separate terms in the momentum conservation equation (Eq. 2.3), and body forces (centrifugal force, Coriolis force, electromagnetic forces and gravitational force), commonly included as source terms. In the following equations, the body forces were not considered since they are not relevant in this frame.

The momentum conservation equation is given by

$$\frac{\partial\left(\rho u_i\right)}{\partial t}+\frac{\partial\left(\rho u_j u_i\right)}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{\partial \tau_{ij}}{\partial x_j} \tag{2.3}$$

where ρ is the density, u_i the i^{th} velocity component, p the pressure and τ_{ij} is the deviatoric stress tensor, which, for generalized Newtonian fluids is given by

$$\tau_{ij}=\eta\left(\dot{\gamma}\right)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{2.4}$$

where the viscosity η, is a function of the shear rate, $\dot{\gamma}$,

$$\dot{\gamma}\equiv\sqrt{2tr\mathbf{D}^2} \tag{2.5}$$

with,

$$D_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{2.6}$$

Several viscosity models can be found in the literature, being the most used models for polymer melts given in Table. 2.1, where η_0 is the zero shear rate viscosity, η_∞ is the infinity shear rate viscosity, λ is a time constant (the inverse of the shear rate at which the fluid changes from Newtonian to shear thinning behavior), n is the power law index, K is the consistency index and α is the Cross Time Constant (the inverse of α gives us a critical shear rate that proves a useful indicator of the onset shear rate for shear thinning). These models account for the effect of the shear rate on the shear viscosity, which reduces with the increase of the shear rate, a phenomenon known as shear thinning.

model	$\eta\left(\dot{\gamma}\right)$
power-law or Ostwale-De Waele	$K\dot{\gamma}^{n-1}$
Carreau	$\left(\eta_0-\eta_\infty\right)\left(1+\left(\lambda\dot{\gamma}\right)^2\right)^{\frac{n-1}{2}}+\eta_\infty$
Carreau simplified	$\eta_0\left(1+\left(\lambda\dot{\gamma}\right)^2\right)^{\frac{n-1}{2}}$
Sisco	$\eta_0+K\dot{\gamma}^{n-1}$
Cross	$\eta_\infty+\frac{\eta_0-\eta_\infty}{1+\alpha\dot{\gamma}^n}$

Table 2.1: Viscosity Models.

The viscosity can also be temperature dependent. To model that dependency, the viscosity is given by Eq. 2.7,

$$\eta\left(\dot{\gamma},T\right)=F\left(\dot{\gamma}\cdot H\left(T\right)\right)F\left(T\right) \tag{2.7}$$

with F being one of the viscous function given in Table 2.1, and H given by Arrhenius law (Eq. 2.8),

$$H\left(T\right)=exp\left(\frac{E}{R}\left(\frac{1}{T}-\frac{1}{T_{ref}}\right)\right) \tag{2.8}$$

where E is the activation energy, R the universal gas constant and T_{ref} the reference temperature (in Kelvin) for which $H(T)=1$.

Energy Conservation Equation

The energy conservation equation, derived from the "First Law of Thermodynamics", states that "the rate of change of energy of a fluid particle is equal to the rate of heat addition to the fluid particle plus the rate of work done on the particle" (Versteeg and Malalasekera, 2007). The energy of a fluid is the sum of internal (thermal) energy i, kinetic energy and gravitational potential energy, hence the internal (temperature) energy equation can be obtained by subtracting the mechanical (kinetic and potential) energy contribution from the total energy. For the case of incompressible fluids, the internal (thermal) energy i, is given by $i = cT$, where c is the specific heat and T the temperature. The energy conservation equation is then given by,

$$\frac{\partial\left(\rho c T\right)}{\partial t} + \frac{\partial\left(\rho c u_j T\right)}{\partial x_j} - \frac{\partial}{\partial x_j}\left(k\frac{\partial T}{\partial x_j}\right) = S_u \tag{2.9}$$

where u is the velocity, k the thermal conductivity and S_u is the source term. If the viscous dissipation is taken into account, the source term is defined by,

$$S_u = \eta\left(\dot{\gamma}\right) D_{ij}\tau_{ij} \tag{2.10}$$

where D_{ij} is the second invariant of stress tensor and τ_{ij} is the deviatoric stress tensor (Eq. 2.4).

General Differential Conservation Equation

The Mass Conservation Equation (Eq. 2.1), the Momentum Conservation Equation (Eq. 2.3) and the Energy Conservation Equation (Eq. 2.9), can all be written as a General Differential Conservation Equation for property ϕ given by,

$$\frac{\partial\left(\rho\phi\right)}{\partial t} + \frac{\partial\left(\rho u_j \phi\right)}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\Gamma\frac{\partial \phi}{\partial x_j}\right) = S_u \tag{2.11}$$

where Γ is the diffusion coefficient and S_u is the source term of property ϕ.

This equation comprises four terms, the *unsteady term*, the *advective term*, the *diffusive term* and the *source term*.

2.2 BOUNDARY CONDITIONS

In problems modeled by differential equations, as those of CFD, in addition to the governing equations, boundary conditions are needed. In the case of transient problems, the initial values in all domain are also required, but in the stationary problems the initial values are only required as initial guesses.

Fig. 2.1 shows a typical CV with a boundary face f and the associated vectors ξ, from CV centre to boundary face centre, and vector n normal to boundary face.

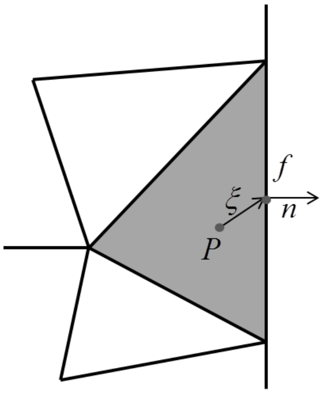


Figure 2.1: Typical CV with a boundary face.

In this case, the boundary conditions considered for the conservation of momentum and continuity equations are:

- inlet: imposed velocity, i. e., $u_i = u_i^0$
- outlet: null normal gradient, i. e., $\frac{\partial u_i}{\partial x_j} n_j = 0$
- wall: no-slip condition, i. e., $u_i = 0$
- symmetry: imposed velocity just with tangential component, i. e., $u_i n_i = 0$

The boundary conditions considered for the Energy Conservation Equation are:

- imposed temperature: $T = T^0$
- insulated and symmetry: $\frac{\partial T}{\partial x_i} n_i = 0$
- natural convection: $k \frac{\partial T}{\partial x_i} n_i = \alpha \left(T_\infty - T_f \right)$, where α is the convection heat transfer coefficient at face f, T_∞ is the environment (far from the surface) temperature and T_f the boundary face temperature
- contact between two domains: $k \frac{\partial T}{\partial x_i} n_i = h_f \left(T_{f_1} - T_{f_2} \right)$, where h_f is the heat transfer coefficient at face f, T_{f_1} and T_{f_2} are the temperature of the contact faces f_1 and f_2, respectively.

2.3 TIME EVOLUTION DISCRETIZATION

After the definition of the mathematical equations that govern the flow and energy transfer, a numerical procedure needs to be set for the numerical solution of the set of equations.

The discretization of equations can be done using different methods, e.g., Finite Difference Method (FDM), Finite Volume Method (FVM) and Finite Element Method (FEM), but, independently of the method applied, the discrete approach obtained should converge to the continuous distribution of the variable, as the number of the space discretization elements increases. In this work the FVM was employed.

In the FVM, the integration of the general governing equation is performed over each control volume and on each time step. By doing so, the general conservation equation (Eq. 2.11) becomes

$$\int\limits_{\Delta V}\int\limits_{\Delta t} \frac{\partial\left(\rho\phi\right)}{\partial t} dt dV + \int\limits_{\Delta t}\int\limits_{\Delta V} \frac{\partial\left(\rho u_j \phi\right)}{\partial x_j} dV dt - \int\limits_{\Delta t}\int\limits_{\Delta V} \frac{\partial}{\partial x_j}\left(\Gamma\frac{\partial\phi}{\partial x_j}\right) dV dt = \int\limits_{\Delta t}\int\limits_{\Delta V} S_\phi dV dt \tag{2.12}$$

where ΔV is the control volume and Δt the time step.

To make the time integration of each term of Eq. 2.12 (except the unsteady term) at a control volume P, it is needed to assume a variation of that term (here represented by Υ_P) between time t and $t + \Delta t$ (Versteeg and Malalasekera, 2007), or write it as a combination of Υ_P value at time t and $t + \Delta t$. In this case, the integral value can be approached by an weighted average of Υ_P and Υ_P^0 values (Eq. 2.13)

$$\int\limits_{\Delta t} \Upsilon_P dt = \left(\theta\Upsilon_P + \left(1-\theta\right)\Upsilon_P^0\right)\Delta t \tag{2.13}$$

where the parameter θ ($\theta \in [0,1]$), defines the weight attributed to the previous (Υ_P^0) and actual (Υ_P) time steps (Fig. 2.2). If $\theta = 0$ only the value of previous time step is considered (Explicit Scheme). Using Explicit Scheme, the contributions to neighbor coefficients are not ensured to be positive, leading to time step restrictions that become very expensive when improving spatial accuracy (Versteeg and Malalasekera, 2007). Additionally, the Explicit Scheme is first-order accurate with respect to time. When $\theta = 1/2$, the method is known as Crank-Nicolson implicit scheme, it is based on central differences and, therefore, is second order accurate in time.

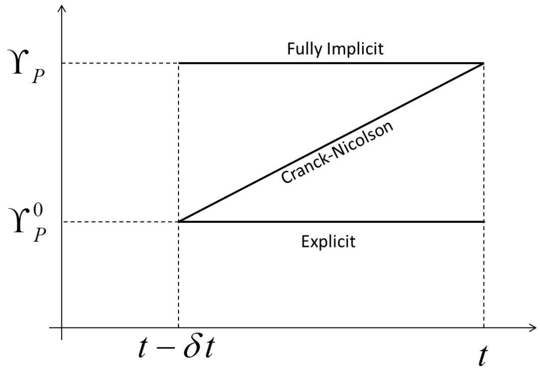


Figure 2.2: Time integration schemes.

If $\theta = 1$ only the current value is considered (Fully Implicit Scheme), all contributions to neighbor coefficients are positive, which makes this scheme unconditionally stable for any time step. The Fully Implicit Scheme is first order accurate in time. Due to its characteristics, this method is recommended for transient calculations, and is employed in this work to study stationary case, being the unsteady term used for convergence purposes. After the time integration, Eq. 2.12 becomes,

$$\int_{\Delta V}\int_{\Delta t} \frac{\partial\left(\rho\phi\right)}{\partial t} dt dV + \int_{\Delta V} \frac{\partial\left(\rho u_j \phi\right)}{\partial x_j} dV\Delta t - \int_{\Delta V} \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) dV\Delta t = \int_{\Delta V} S_\phi dV\Delta t \quad (2.14)$$

The Eq. 2.14 is dependent of a variable ϕ, continuous in time and space and its analytical solutions rarely exist (Tropsa, 2001), being always mathematically very complex. In order to solve this governing differential equation it is replaced by a discretized version (using numerical methods), connecting the values of ϕ for a finite set of points through a set of equations that can be easily solved (Patankar, 1980). Each point in space, obtained by discretization, generates an equation with a general form of Eq. 2.15, and the resulting system of equations can be solved using computers.

$$a_P\phi_P = \sum_{nb} a_{nb}\phi_{nb} + S_u \tag{2.15}$$

2.4 DISCRETIZATION OF THE GENERAL DIFFERENTIAL CONSERVATION EQUATION

To assemble the CV equations (Energy and Momentum Conservation Equations), the contribution of each term is considered independently. Hereafter, the discretization of the governing equation (Eq. 2.14) is made considering singly each of its terms. In Eq. 2.14 the terms appear in the following order: unsteady, advective, diffusive and source terms. This is also the order in which that their discretization will be detailed in the following sections.

In the advective, diffusive and source terms of the momentum conservation equation, gradients needed to be estimated; therefore, the methodology adopted to estimate the gradient of a generic variable ϕ will be explained before describing the discretization procedure.

Gradient computation

To estimate the gradient of a property ϕ, a truncated Taylor series expansion (Eq. 2.16) around the center of the control volume is considered,

$$\phi_{nb} = \phi_P + \left(\frac{\partial\phi}{\partial x_i}\right)_P \xi_i + O\left(\xi_i\xi_i\right) \tag{2.16}$$

where ξ is the vector difference that connects the geometric centre of the CV_P to a neighbor CV_{nb}, $\left(\frac{\partial\phi}{\partial x_i}\right)_P$ is the gradient of ϕ at CV_P and $O\left(\xi_i\xi_i\right)$ is the truncation error.

For the boundary faces of the CV, the ξ vector connects the geometric centres of CV and boundary face, being the value of ϕ in the boundary updated in a specific function, according to its boundary condition.

Since the neglected terms are proportional to the square distance $\xi_i\xi_i$, this is a second-order approximation. To evaluate the gradient the *least squares gradient* *reconstruction* was used. Neglecting the high order accuracy terms, Eq. 2.16 can be rewritten as,

$$\phi_{nb} - \phi_P = \left(\frac{\partial \phi}{\partial x_i}\right)_P \xi_i \tag{2.17}$$

Considering Eq. 2.17 for each neighbor CV_{nb} of the CV_P (Fig. 2.3), an overdetermined system of equations (Eq. 2.18) with the form $Ax = b$, provides the solution, $\left(\frac{\partial \phi}{\partial x_i}\right)_P$.

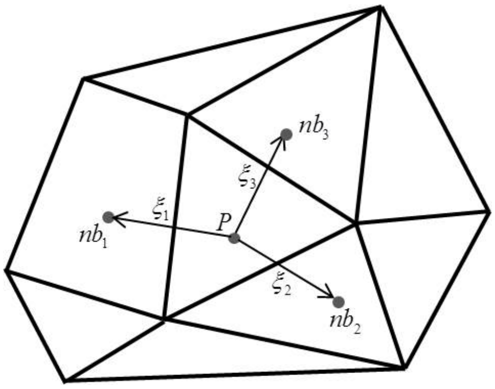


Figure 2.3: Cell arrangement considered for the computation of the gradient approximation.

$$\begin{bmatrix} x_{nb,1} - x_P & y_{nb,1} - y_P & z_{nb,1} - z_P \\ x_{nb,2} - x_P & y_{nb,2} - y_P & z_{nb,2} - z_P \\ \vdots & \vdots & \vdots \\ x_{nb,n} - x_P & y_{nb,n} - y_P & z_{nb,n} - z_P \end{bmatrix} \begin{bmatrix} \left(\frac{\partial \phi}{\partial x}\right)_P \\ \left(\frac{\partial \phi}{\partial y}\right)_P \\ \left(\frac{\partial \phi}{\partial z}\right)_P \end{bmatrix} = \begin{bmatrix} \phi_{nb,1} - \phi_P \\ \phi_{nb,2} - \phi_P \\ \vdots \\ \phi_{nb,n} - \phi_P \end{bmatrix} \tag{2.18}$$

Multiplying both sides of the equation by the transpose A^T, we obtain a 3×3 matrix $A^T A$ on the right-hand-side of Eq. 2.19, whose inverse can be easily obtained allowing the solution of the equation (Eq. 2.20).

$$A^T A x = A^T b \tag{2.19}$$

$$x = \left(A^T A\right)^{-1} A^T b \tag{2.20}$$

Since $\left(A^TA\right)^{-1}A^T$ is only dependent on the geometry, it can be evaluated only once at the beginning of the calculations for each CV.

Unsteady term

In this work only steady problems were considered, nevertheless the unsteady term was used by relaxation purposes.

The unsteady term is treated as constant in space, hence the volume integration is straightforward (Eq. 2.21).

$$\int\limits_{\Delta V}\int\limits_{\Delta t}\frac{\partial\left(\rho\phi\right)}{\partial t}dtdV=\Delta V\int\limits_{\Delta t}\frac{\partial\left(\rho\phi\right)}{\partial t}dt \tag{2.21}$$

The time integration, results in the following equation,

$$\Delta V\int\limits_{\Delta t}\frac{\partial\left(\rho\phi\right)}{\partial t}dt=\underbrace{\Delta V\rho}_{a_P}\phi_P-\underbrace{\Delta V\rho\phi_P^0}_{S_u}$$

leading to a contribution to the diagonal (a_P) of the system of equations, and a contribution to the source term (S_u) in Eq. 2.15.

Advective term

The integration of the advective term, after applying the Gauss Divergence Theorem, results in a surface integral over the control volume surface,

$$\int\limits_{\Delta V}\frac{\partial\left(\rho u_j\phi\right)}{\partial x_j}dV\Delta t=\int\limits_{\partial V}\left(\rho u_j\phi\right)n_jdA\Delta t \tag{2.22}$$

where ∂V is the surface of the CV.

The control volumes (CV) are polyhedrons (Fig. 2.4), hence the surface integral can be substituted by a sum of integrals in each CV face,

$$\int\limits_{\partial V}\left(\rho u_j\phi\right)n_jdA\Delta t=\sum_f\int\limits_{A_f}\left(\rho u_j\phi\right)n_jdA\Delta t. \tag{2.23}$$

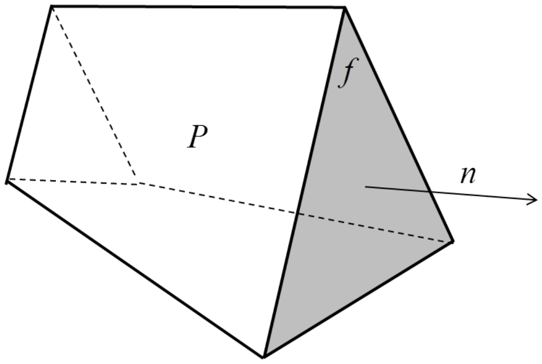


Figure 2.4: CV (polyhedral shape) with the normal vector of one of its faces.

To evaluate the area integral we use the mean value theorem. The values of ϕ at the surface of the control volume (ϕ_f) are obtained by interpolation from the neighbors values. After some rearrangement, the right-hand side of Eq. 2.23 is transformed (Eq. 2.24),

$$\sum_f \int_{A_f} \left(\rho u_j \phi\right) n_j dA\Delta t = \sum_f A_f \left(u_j n_j\right)_f \rho \Delta t \phi_f \tag{2.24}$$

and the normal component of velocity $\left(u_j n_j\right)_f$ is treated using other procedure, to assure coupling between velocity and pressure fields, that will be explained later. The value of ϕ at the surface, can be written depending on the neighbor CV variables. In Central Differencing Scheme (CDS), the weight of both CV is the same, but on strongly convective flows, the face value (ϕ_f) receives a greater influence from the upwind cell value. The Upwind Differencing Scheme (UDS) takes into account the flow direction and approaches the face value (ϕ_f) by its upwind cell value. The CDS can produce solutions that oscillate about the exact solution, whereas the UDS causes erroneous results when the flow is not aligned with the grid lines (false diffusion). Since UDS is more stable and CDS have higher-order accuracy, a better choice should be a combination of both, which can be written as,

$$\phi_f = \phi_U + \frac{1}{2}\psi\left(r\right)\left(\phi_D - \phi_U\right) \tag{2.25}$$

where the subscript U stands for its upwind and D to its downwind neighbor. This generalization, that uses a limiter function, $\psi\left(r\right)$ (Fig. 2.5), is called Total Variation

Diminishing (TVD) and has been specially formulated to achieve oscillation-free solutions that result on higher order schemes (Versteeg and Malalasekera, 2007).

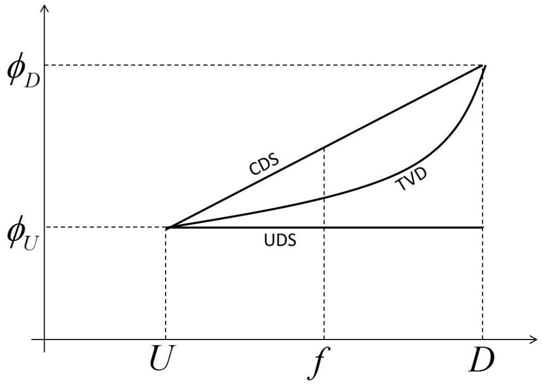


Figure 2.5: Total Variation Diminishing schemes.

The limiter function $\psi(r)$ is connected to the gradient of the transported quantity ϕ, at the surface of the control volume. In structured meshes the parameter r is given by Eq. 2.26, for a control volume P.

$$r = \frac{\phi_P - \phi_U}{\phi_D - \phi_P}. \tag{2.26}$$

In Fig. 2.6 we can see an example where $r < 1$ ($\phi_P - \phi_U < \phi_D - \phi_P$). The values obtained to ϕ_f using CDS or using UDS are far from those suggested by an interpolation scheme, using the values of ϕ_U, ϕ_P and ϕ_D with higher order than one. Such interpolation scheme suggests that ϕ_f should belong to an interval limited by the estimates obtained by UDS and CDS, i.e., $\psi(r)$ should belong to $]0, 1[$ (Eq. 2.25).

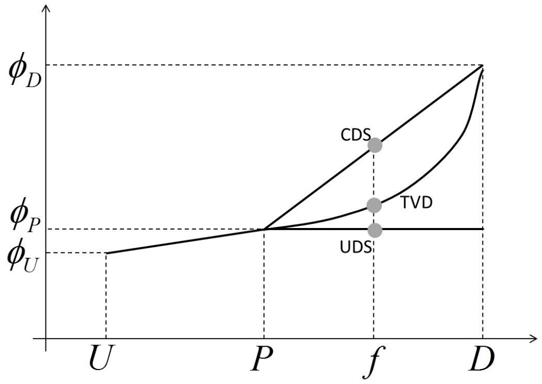


Figure 2.6: Face value evaluation with a TVD scheme

Sweby (1984) established the conditions to the limiter function, in order to have a stable and a non-oscillatory higher-order scheme and second order accuracy (shaded region in Fig. 2.7). Over the years, a number of limiter functions that satisfy Sweeby's requirements have been developed and successfully used (Sweby, 1984).

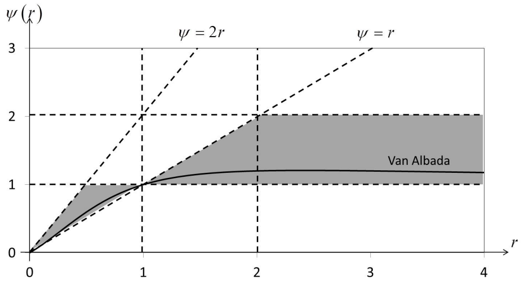


Figure 2.7: Region for a second-order TVD scheme.

As an example, the limiter function suggested by Van Albada (Eq. 2.27) produces better results (smaller false diffusion) than UDS (Versteeg and Malalasekera, 2007),

$$\psi\left(r\right) = \frac{r + r^2}{1 + r^2} \tag{2.27}$$

In the case of unstructured meshes the parameter r cannot be calculated in the same way as it is in the structured ones, because the upwind CV is not available. For unstructured meshes, a "dummy" upwind CV must be constructed to obtain a similar formula. Darwish and Moukalled (2003) recommended Eq. 2.28 to calculate r in case of unstructured meshes,

$$r = \frac{2\left(\overline{\frac{\partial \phi}{\partial x_i}}\right)_U \xi_i}{\phi_D^0 - \phi_U^0} - 1 \tag{2.28}$$

where U is the CV upwind of the face and D the downwind CV, and the estimate to gradient $\left(\overline{\frac{\partial \phi}{\partial x_i}}\right)_U$ is calculated with values from the neighbors of cell U, as explained before (see Gradient computation).

Substituting Eq. 2.25 on Eq. 2.24, after replacing ϕ_U and ϕ_D with the previous iteration values ϕ_P^0 and ϕ_{nb}^0 taking into account the flow direction, we obtain the discretization of advective term (Eq. 2.29) with contributions to the central coefficient of

CV_P (a_P) (matrix diagonal), or contributions to the central coefficient of its neighbors (a_{nb}) and to the source term (S_u) of the CV_P equation (Eq. 2.29).

$$\Delta t A_f \left(u_j n_j\right)_f \rho \left(\phi_U + \frac{\psi\left(r\right)}{2}\left(\phi_D^0 - \phi_U^0\right)\right) \tag{2.29}$$

If the normal component of velocity and the vector normal to the control volume surface have the same direction, the upwind CV (U) and the downwind CV (D) are replaced by P and nb, respectively (Eq. 2.31).

$$\Delta t A_f \left(u_j n_j\right)_f \rho \left(\phi_P + \tfrac{\psi(r)}{2}\left(\phi_{nb}^0 - \phi_P^0\right)\right) \tag{2.30}$$

$$= \underbrace{\Delta t A_f \left(u_j n_j\right)_f \rho \phi_P}_{a_P} + \underbrace{\Delta t A_f \left(u_j n_j\right)_f \rho \frac{\psi\left(r\right)}{2}\left(\phi_{nb}^0 - \phi_P^0\right)}_{S_u} \tag{2.31}$$

If the normal component of velocity is opposite to the normal surface direction, the upwind CV (U) and the downwind CV (D) are replaced by nb and P, respectively (Eq. 2.33).

$$\Delta t A_f \left(u_j n_j\right)_f \rho \left(\phi_{nb} + \tfrac{\psi(r)}{2}\left(\phi_P^0 - \phi_{nb}^0\right)\right) \tag{2.32}$$

$$= \underbrace{\Delta t A_f \left(u_j n_j\right)_f \rho \phi_{nb}}_{a_{nb}} + \underbrace{\Delta t A_f \left(u_j n_j\right)_f \rho \frac{\psi\left(r\right)}{2}\left(\phi_P^0 - \phi_{nb}^0\right)}_{S_u} \tag{2.33}$$

Diffusive term

As was done for the advective term, the volume integral of the diffusive term is transformed into a surface integral (Eq. 2.34), through the employment of the Gauss Divergence Theorem,

$$\int\limits_{\Delta V} \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) dV \Delta t = \int\limits_{\partial V} \Gamma \frac{\partial \phi}{\partial x_j} n_j dA \Delta t \tag{2.34}$$

that can be written as a sum of integrals in each of CV faces (Eq. 2.35).

$$\int_{\partial V} \Gamma \frac{\partial \phi}{\partial x_j} n_j dA\Delta t = \sum_f \Gamma \left(\frac{\partial \phi}{\partial x_j} \right)_f n_{f,j} A_f \Delta t \tag{2.35}$$

In orthogonal meshes the variation of ϕ along the normal direction (n) can be approximated using central differences, since the vector normal to the face (n) and the vector that links the two CV centres (ξ) are coincident. However, in unstructured meshes the vectors n and ξ are not necessarily coincident (see Fig. 2.8) and that must be taken into account.

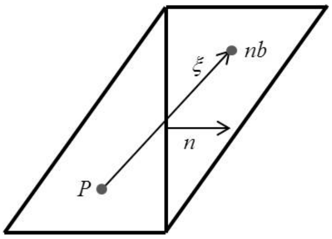


Figure 2.8: Typical mesh arrangement in unstructured meshes.

This is usually referred as skewness or non-orthogonality of the mesh. To account for this situation, a new term, called "cross-diffusion" is added that is calculated with values obtained on previous iterations. In this work we use the following equation to approximate the diffusive term:

$$\sum_f \Delta t A_f \Gamma \left(\left(\frac{\partial \phi}{\partial x_k} \right)_f + \left(\frac{\phi_{nb} - \phi_P}{||\xi||} - \frac{\left(\overline{\frac{\partial \phi}{\partial x_j}} \right)_f \xi_j}{||\xi||} \right) \frac{||\xi|| n_k}{\xi_l n_l} \right) n_k \tag{2.36}$$

that was devised by Muzaferija (Bašić et al., 2005), where the second term (inside the outer brackets) is a third-order artificial diffusion introduced to smooth the unphysical oscillations caused by odd-even decoupling present in methods that use a collocated variable arrangement. The expression $\left(\frac{\partial \phi}{\partial x_j} \right)_f$ is evaluated (as explained in Gradient computation) using the neighbors values of ϕ (that share the face f) and $\left(\overline{\frac{\partial \phi}{\partial x_j}} \right)_f$ is the mean value of the estimated gradients on both CV that share face f.

The diffusive term contribution (Eq, 2.36) can be rearranged into Eq. 2.37

$$\sum_f \left(\Delta t A_f \Gamma \left(\frac{\partial \phi}{\partial x_k} \right)_f n_k + \Delta t A_f \Gamma \left(\frac{\phi_{nb} - \phi_P}{\|\xi\|} - \frac{\left(\overline{\frac{\partial \phi}{\partial x_j}} \right)_f \xi_j}{\|\xi\|} \right) \frac{\|\xi\| n_k n_k}{\xi_l n_l} \right) \tag{2.37}$$

and, considering that $n_k n_k = 1$ and $\|\xi\| / \|\xi\| = 1$, the contribution of each face f is transformed into Eq. 2.38 in order to highlight the contributions to each coefficient of CV P equation.

$$\underbrace{\frac{\Delta t A_f \Gamma}{\xi_l n_l}}_{a_{nb}} \phi_{nb} - \underbrace{\frac{\Delta t A_f \Gamma}{\xi_l n_l}}_{a_P} \phi_P + \underbrace{\Delta t A_f \Gamma \left(\left(\frac{\partial \phi}{\partial x_j} \right)_f n_j - \frac{\left(\overline{\frac{\partial \phi}{\partial x_j}} \right)_f \xi_j}{\xi_l n_l} \right)}_{S_u} \tag{2.38}$$

Source term

The discretization of the source term is given by

$$\int_{\Delta t} \int_{\delta V} S_\phi dV dt = \underbrace{\Delta V \Delta t S_\phi}_{S_u} \tag{2.39}$$

Boundary conditions implementation

For boundary faces with ϕ imposed (imposed temperature and inlet boundaries), the advective term discretization is straightforward, since the value at face ϕ_f is known. For the case of the diffusive term, when ϕ is imposed at the boundary, it is calculated with a first order approximation to the gradient along the face normal direction,

$$\frac{\partial \phi}{\partial x_j} n_j \simeq \frac{\phi_f - \phi_P}{\xi_j n_j} \tag{2.40}$$

where ϕ_P is the property ϕ of the CV that contains the boundary face, and ξ the vector that connects the geometric centre of the CV and the geometric centre of the boundary face (see Fig. 2.9).

For zero flux boundary faces (insulated for energy equation and outlet for momentum equations), the approximated diffusive term is obtained by simple substitution

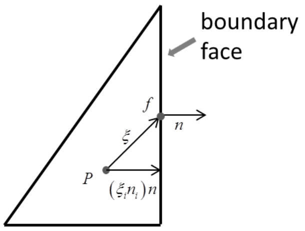


Figure 2.9: Typical configuration of a cell that has a boundary face

of $\frac{\partial \phi}{\partial x_j} n_j$ by 0. For the advective term, ϕ_f is approached by the value at the CV (ϕ_P) as suggest by Eq. 2.40 when $\frac{\partial \phi}{\partial x_j} n_j = 0$.

At the symmetry boundary faces, there is no advection neither diffusion, but the boundary variables must be updated since their values are used to compute the gradients. The boundary face velocity is defined as the tangential component of the respective CV velocity, which should have a null normal component. Thus the boundary velocity is given by Eq. 2.41.

$$u_{f,i} = u_{P,i} - u_{P,j} n_j \tag{2.41}$$

To update the temperature at boundary faces in contact with other domain, or in contact with air (natural convection), Eq. 2.42 is used, considering the respective CV and estimate of its gradient (see Gradient computation). In insulated faces the temperature is updated with its element temperature since there is no variation in the normal direction. In the boundary faces with imposed temperature there is no need to update its temperature.

$$\phi_f = \phi_P + \left(\overline{\frac{\partial \phi}{\partial x_j}} \right)_P n_j \left(\xi_k n_k \right) \tag{2.42}$$

The pressure at the boundary faces have to be updated, in order to enable the evaluation of pressure gradients at the respective CV. The Eq. 2.42 is used to update the pressure at the inlet and outlet boundary faces, thus a second order interpolation scheme is employed. On boundary faces with symmetry boundary condition or wall, the pressure equals the value of the respective CV, thus a null gradient is assumed on that direction.

2.5 CALCULATION OF THE VELOCITY AND PRESSURE FIELDS

In flow field problems, the unknown velocity field could be obtained by Momentum Conservation Equations (Eq. 2.3) for each of velocity components u_i, if the pressure gradient was known. However, generally, this gradient is part of the problem unknowns. Furthermore, the momentum conservation equation is non-linear since the advective term has a multiplication of two velocity components. In flow field problems we have four unknowns, namely, pressure and three velocity components, but there is no equation to obtain directly the pressure, despite the four available equations (three momentum conservation equations and the mass conservation equation). If the pressure field was known, the non-linearity of the advective term could be solved with an iterative process, where the velocity at the inner faces was considered to be known, to get the velocity at the CV centres and, then, the CV velocities are used to update the inner face velocities, by interpolation. The process should be repeated in an iterative fashion until convergence is achieved.

To solve the above mentioned problems, the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm presented by Patankar and Spalding (1972), that allow to calculate the pressure, was implemented. The SIMPLE algorithm starts by assuming a initial guessed values for pressure (p^*), velocity at faces $(u_i n_i)_f$, and other variables, e. g., temperature or viscosity, that can influence the flow distribution.

With the initial estimates, the momentum conservation equations are assembled and solved to obtain the velocity components (u_i^*) at the CV centres.

With the new estimate for velocity at CV, the boundary faces velocities are updated according to their boundary type and, additionally, the velocities at the outlet boundary faces are multiplied by a factor in order to ensure global mass conservation, i. e., the outlet flow rate should be equal to the inlet flow rate.

To avoid decoupling between pressure and velocity fields, the dot product $(u_i n_i)_f$ on advective term is computed using Rhie and Chow (Versteeg and Malalasekera, 2007) interpolation scheme (Eq. 2.43),

$$\begin{aligned}(u_i n_i)_f = & \quad \tfrac{u^*_{i,P}+u^*_{i,nb}}{2} n_{i,f} \\ + & \quad \tfrac{1}{2}\left(\tfrac{\Delta V_P}{a_P}+\tfrac{\Delta V_{nb}}{a_{nb}}\right)\tfrac{p_P-p_{nb}}{\|\xi\|} \\ - & \quad \tfrac{1}{2}\left(\tfrac{\Delta V_P}{a_P}\left(\overline{\tfrac{\partial p}{\partial x_i}}\right)_P+\tfrac{\Delta V_{nb}}{a_{nb}}\left(\overline{\tfrac{\partial p}{\partial x_i}}\right)_{nb}\right)\tfrac{\xi_{i,f}}{\|\xi\|}\end{aligned} \tag{2.43}$$

where the pressure gradients $\left(\overline{\frac{\partial p}{\partial x_i}}\right)$ are obtained by least square gradients (see Gradientes computation), ΔV_P and ΔV_{nb} are the volumes of the CV_P and CV_{nb}, respectively, and a_P and a_{nb} the diagonal coefficients of momentum conservation equation system of CV_P equation and CV_{nb} equation, respectively.

At this stage, velocity and pressure fields verify the momentum conservation equations but mass is not conserved.

To ensure mass conservation at each CV, the faces velocities must be corrected. Therefore, following the usual Finite Volume procedure, the mass conservation equation (Eq. 2.2) is integrated over each control volume and the velocity at faces u_i are substituted by the guessed velocity (u^*_f) plus an unknown correction $(u^{'}_f)$ (Eq. 2.44).

$$\sum_f A_f\left((u^*_i n_i)_f + (u'_i n_i)_f\right) = 0 \tag{2.44}$$

Subtracting Eq. 2.43 for the correct faces velocities (u_f) by a similar equation with guessed velocities (u^*_f), a equation for the face velocities corrections is obtained (Eq. 2.45).

$$(u^{'}_i n_i)_f = \quad \tfrac{u^{'}_{i,P}+u^{'}_{i,nb}}{2} n_{i,f} + \tfrac{1}{2}\left(\tfrac{\Delta V_P}{a_P}+\tfrac{\Delta V_{nb}}{a_{nb}}\right)\tfrac{p^{'}_P-p^{'}_{nb}}{\|\xi\|} \tag{2.45}$$

Neglecting the contributions ($u^{'}_{i,P}$ and $u^{'}_{i,nb}$) of CV that share the face (main approach of SIMPLE (Versteeg and Malalasekera, 2007)) a new equation to the face velocity correction is obtained (Eq. 2.46).

$$(u'_i n_i)_f = \frac{1}{2}\left(\frac{\Delta V_P}{a_P}+\frac{\Delta V_{nb}}{a_{nb}}\right)\frac{p'_P - p'_{nb}}{\|\xi\|} \tag{2.46}$$

Substituting the expression of velocity correction given by Eq. 2.46 on Eq. 2.44, one obtains a pressure correction equation (Eq. 2.47), for each CV_P,

$$\underbrace{\sum_f \frac{A_f}{2\|\xi\|}\left(\frac{\Delta V_P}{a_P}+\frac{\Delta V_{nb}}{a_{nb}}\right)}_{a'_P} p'_P - \sum_f \underbrace{\frac{A_f}{2\|\xi\|}\left(\frac{\Delta V_P}{a_P}+\frac{\Delta V_{nb}}{a_{nb}}\right)}_{a'_{nb}} p'_{nb} = \underbrace{-\sum_f A_f (u_i^* n_i)_f}_{S'_u} \tag{2.47}$$

where a'_P is the coefficient of p', a'_{nb} are the coefficients of $p^{'}_{nb}$ and S'_P is the right-hand-side of the pressure correction equation for each CV, that can be formulated for all the CV, and thus a system of equations that allow computing the pressure correction field is obtained.

After solving the system of equations, and thus computing the pressure correction field, the guessed pressure on CV (p^*_P) can be corrected. To avoid divergence, the pressure correction should be under-relaxed ($p_P = p^*_P + \alpha p'_P$, with $\alpha = 0.1$) (Versteeg and Malalasekera, 2007). The face velocities can be also corrected using the pressure correction field (p') on Eq. 2.46. The computed fluxes (or normal face velocities) generates a mass conservative field on each CV.

To correct the velocities on CV, a new expression is required to update the CV velocities. For that purpose, the momentum conservation equations for the guessed and exact velocity and pressure fields, given by,

$$a_P u_{P,i} = \sum_{nb} a_{nb} u_{nb,i} - \frac{\partial p}{\partial x_i}\Delta V_P \tag{2.48}$$

$$a_P u^*_{P,i} = \sum_{nb} a_{nb} u^*_{nb,i} - \frac{\partial p^*}{\partial x_i}\Delta V_P \tag{2.49}$$

are subtracted leading to the required equation:

$$a_P u'_{P,i} = \sum_{nb} a_{nb} u'_{nb,i} - \frac{\partial p'}{\partial x_i}\Delta V_P \tag{2.50}$$

As in the main approach of SIMPLE, the neighbors contributions ($a_{nb}u'_{nb,i}$) are neglected and an explicit expression for the velocity correction field at the CV centres is obtained (Eq. 2.51),

$$u'_{P,i} = -\frac{\Delta V_P}{a_P}\frac{\partial p'}{\partial x_i} \tag{2.51}$$

where the value of pressure derivative $\frac{\partial p'}{\partial x_i}$ is estimated with known neighbor CV values and using the least squares method $\left(\frac{\partial p'}{\partial x_i} \simeq \left(\overline{\frac{\partial p'}{\partial x_i}}\right)_P\right)$ (see Section 2.4).

At this stage, the convergence is evaluated by the initial error of momentum conservation equations $\left(\sqrt{\sum\limits_{cells}\left(a_P\phi_P - \sum\limits_{nb} a_{nb}\phi_{nb} - S_u\right)^2}\right)$ system and by mass conservation residuals. If no convergence is achieved, the process proceeds (Fig. 2.10) to next time step with the current values as new estimates to pressure and velocities on CV, and fluxes at the CV faces.

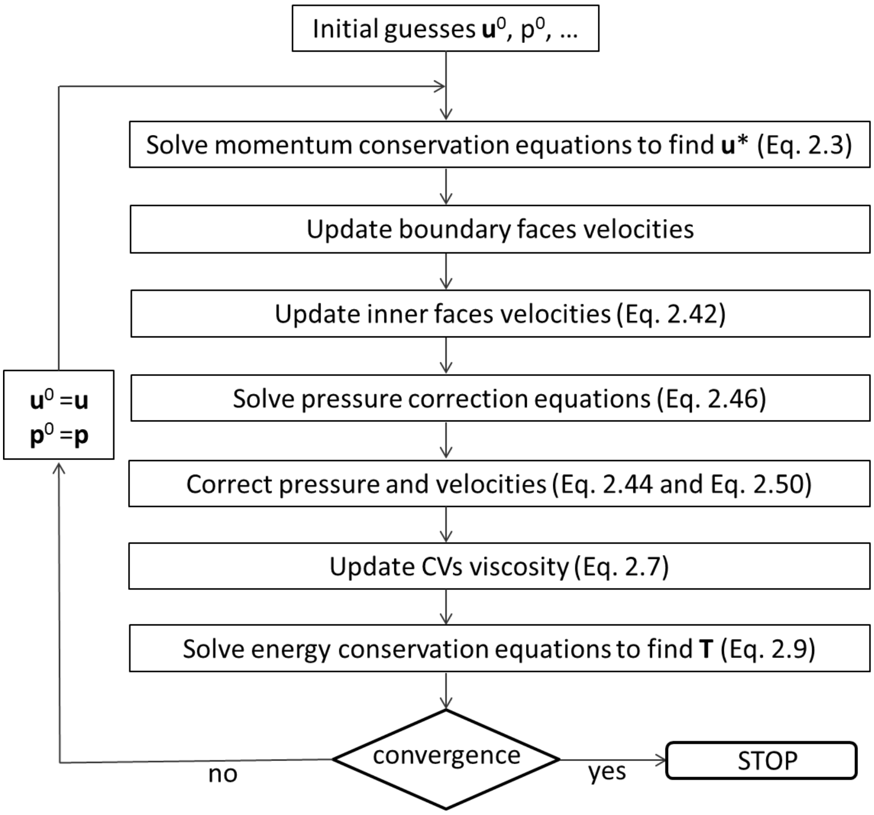

Initial guesses **u**⁰, p⁰, ...
Solve momentum conservation equations to find **u*** (Eq. 2.3)
Update boundary faces velocities
Update inner faces velocities (Eq. 2.42)
Solve pressure correction equations (Eq. 2.46)
Correct pressure and velocities (Eq. 2.44 and Eq. 2.50)
Update CVs viscosity (Eq. 2.7)
Solve energy conservation equations to find **T** (Eq. 2.9)
convergence
no
yes
STOP
u⁰ = **u**
p⁰ = **p**

Figure 2.10: The SIMPLE algorithm flowchart.

3

DESIGN OF COMPLEX PROFILE EXTRUSION DIES THROUGH NUMERICAL MODELING

Abstract[1]

The achievement of a balanced flow is one of the major tasks encompassed in the design of profile extrusion dies (Hristov and Vlachopoulos, 2007). For this purpose numerical modeling codes may be a very useful aid. The research team involved in this work has been working during the last decade on the development of numerical tools to aid the conception of extrusion dies (Nóbrega, Carneiro, Pinho and Oliveira, 2004). The design code developed so far carries out the automatic search of a final geometry via an optimization routine coupled with geometry and mesh generators and a 3D computational fluid dynamics (CFD) code based on the finite volume method (FVM). This CFD code is able to model the flow of polymer melts in confined channels, but is inadequate to deal with complex geometries, since it is limited to structured meshes. This work describes the recent efforts made to enlarge the scope of the design procedures, that are currently focused on the development of a modeling code able to deal with unstructured meshes. This code solves the continuity and linear momentum conservation equations, with generalized Newtonian fluids, using a SIMPLE based approach. This work describes the developed numerical modelling code and its employment in a case study that involves the design of a medical catheter extrusion die, focused on the search of a balanced flow distribution. The results obtained show that the developed numerical code is able to deal with complex geometrical problems, being thus a valuable tool to aid the design of extrusion dies to produce complex profiles.

[1] N. D. Gonçalves, O. S. Carneiro and J. M. Nóbrega (2013). Design of Complex Profile Extrusion Dies Through Numerical Modeling, Journal of Non-Newtonian Fluid Mechanics, Volume 200, October 2013, 103-110.

3.1 INTRODUCTION

Polymer extrusion is a processing technique widely used in thermoplastic industry to produce parts for medical, building, automotive, electric and electronic applications, among others, due to its ability to generate almost any cross section geometry profiles.

The design of a new product and corresponding extrusion die, must take into account several issues namely, achievement of balanced flow (Nóbrega, Carneiro, Pinho and Oliveira, 2004, 2003; Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004), avoidance of rheological defects, maximization of the production rate (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004; Nóbrega et al., 2008; Carneiro et al., 2001) and maintenance, or increase, of the product quality (Hristov and Vlachopoulos, 2007). Profiles like window frames and catheters, are some of the applications that involve the most complex cross section geometries produced by extrusion. Designing extrusion dies for the production of such geometries, requires a deep knowledge, which is usually based on experimental trial-and-error approaches, involving, therefore, the use of huge amounts of time and material resources (Brown et al., 1979; Schut, 2003). For these reasons, the extrusion die design process can become too difficult to execute, or its cost can increase up to prohibitive levels, when complex geometry thermoplastic profiles are concerned (Michaeli, 1992; Tadmor and Cogos, 2006). To assist in the design of extrusion dies, there are some commercially available softwares, e.g., *ANSYS Polyflow*, *Dieflow*, *HyperXtrude*, *FLOW 2000* and *PROFILECAD*; some of them are even able to solve the so called inverse problem (Legat and Marchal, 1992, 1993; Marchal and Goublomme, 2000), i.e., to compute the flow-channel cross section required to produce a specific profile. Despite all the available tools, the main decisions are left to the designer (Shahreza et al., 2010), requiring, therefore, human intervention and knowledge. In order to automatize the extrusion die design process, some numerical codes have been developed, aiming to transfer much of those decisions to the code, e.g., the work of Ettinger et al. (2004), where the main parameters and optimization strategies required to automatize the design of profile extrusion dies are identified. However, this is restricted to 2D approaches, which limits its applicability. According to the authors knowledge, this methodology had not been further developed. Other advances on automatic optimization of profile extrusion dies have been made by Nóbrega, Carneiro, Pinho and Oliveira (2003); Carneiro et al. (2001); Carneiro, Nóbrega, Oliveira and Pinho (2003) using several strategies, namely those encompassing: changes performed in the flow channel parallel zone cross section or die land, or those involving modifications performed in upstream regions. Despite of the promising results achieved, the numerical code used for this purpose is limited to simple geometry problems, inherent to the use of structured meshes. In order to extend this work to complex geometries, the use of unstructured meshes (Gonçalves, 2007) is mandatory.

The Finite Volume Method (FVM) (Ferziger and Peric, 2002) is a discretization method that can accommodate any type of grid, being, therefore, suitable to deal with complex geometries. The grid defines only the control volume (CV) boundaries and need not to be related to a coordinate system. The method is conservative by construction, as long as surface integrals (which represent convective and diffusive fluxes) are the same for the CVs sharing a common face. The FVM approach is perhaps the simplest to understand and to implement, since all terms that need to be approximated have physical meaning, fact that justifies its popularity among engineers. In addition to the discretization method, one has to select the type of grid that will discretize the problem domain. Finite Volume Method can be used with structured, block-structured or unstructured grids. The disadvantages of structured grids are its lack of adequacy for complex solution domains and the low control over the distribution of the grid points, in locations where the grid has to be refined for accuracy reasons (Ferziger and Peric, 2002). The disadvantage of block-structured grids is that the interfaces of adjacent blocks must be treated in a fully conservative manner, which, in some cases, involve additional iterative procedures to unify the boundary conditions (Ferziger and Peric, 2002). For very complex geometries unstructured grids are more flexible than structured grids or block-structured ones (Versteeg and Malalasekera, 2007). The elements, or control volumes, may have any shape and there is no restriction concerning the number of adjacent cells meeting at a point (2D) or along a line (3D). In practice, grids made of triangles or quadrilaterals in 2D, and tetrahedrons or hexahedrons in 3D, are most often used, and combinations of different type of elements can be made in the same grid. Such grids can be automatically generated and, if desired, can be orthogonal, or locally refined. The disadvantage of unstructured grids is the irregularity of data structure, that can increase the computation time required: since the matrix of algebraic equations system is irregular, the solvers are usually slower than those corresponding to regular ones. On the other hand, computer codes for unstructured grids are more flexible. They need not to be changed when the grid is locally refined, or when elements of different shapes are used. The most attractive feature of the unstructured meshes is that they allow the calculation of flows in or around geometrical features of arbitrary complexity without requiring a long time spent on mesh generation and mapping. In unstructured meshes, two different ways of defining the control volumes are possible (Ferziger and Peric, 2002; Versteeg and Malalasekera, 2007; Blazek, 2006): in the cell-centred method the unknowns are placed at the centroid of the control volume; in the vertex-centred method they are placed on the vertices of the grid, followed by a process known as median-dual tessellation, whereby sub-volumes join centroids of the elements and midpoints of the edges. The sub-volume surrounding a node forms the control volume for discretization. Both cell-centred and vertex-centred methods are used in practice, but cell-centred method is simpler to understand and to implement, and, since a control volume always has more vertices than centroids, it has slightly lower storage requirements than the vertex-centred method.

The objective of this work is to describe a numerical code able to solve flow problems involving complex geometries, since it uses unstructured meshes, to verify it and to illustrate its use on the design of a complex extrusion die required for the production of a medical catheter. Here, the optimization of the design was done manually, i.e., it required the user intervention in the generation of the successive trials. However, the goal in the near future is to automatize this process (Shahreza et al., 2010; Ettinger et al., 2004; Sienz et al., 2010).

In the first part of this work (Sections 3.2 and 3.3), the conservation equations and its discretization are presented. In Section 3.4 the implemented numerical code is verified. In Section 3.5 the code is used to optimize the flow balance of an extrusion die for the production of a medical catheter comprising five lumens. To reach an acceptable flow distribution at the die exit, several geometry corrections were made. Finally, in Section 3.6, the main conclusions of the work are drawn.

3.2 GOVERNING EQUATIONS

The Navier Stokes equations were used to model the flow with an incompressible and generalized Newtonian fluid under isothermal conditions. Taking this into account, the governing equations are the mass conservation:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.1}$$

and the i^{th} momentum conservation equation:

$$\frac{\partial \left(\rho u_i\right)}{\partial t} + \frac{\partial \left(\rho u_j u_i\right)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_i} \tag{3.2}$$

where u_i is the i^{th} velocity component, ρ the density, p the pressure and τ_{ij} is the deviatoric stress tensor, which for generalized Newtonian fluid is given by:

$$\tau_{ij} = \eta\left(\dot{\gamma}\right)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{3.3}$$

where the shear rate, $\dot{\gamma}$, is a function of the second invariant of the rate of deformation tensor:

$$\dot{\gamma} \equiv \sqrt{2tr\mathbf{D}^2} \tag{3.4}$$

$$D_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{3.5}$$

A Bird-Carreau constitutive equation was employed to model the shear viscosity dependence on shear rate:

$$\eta\left(\dot{\gamma}\right) = \eta_0 + \frac{\eta_0 - \eta_\infty}{\left(1 + \left(\lambda\dot{\gamma}\right)^2\right)^{\frac{1-n}{2}}} \tag{3.6}$$

being η_0 the zero shear-rate viscosity, η_∞ the viscosity at very high shear rates, λ a characteristic time and n the power-law index.

Due to the explicit nature of the constitutive equation (Eq. 3.3) it can be directly inserted in the momentum conservation equations, giving:

$$\frac{\partial\left(\rho u_i\right)}{\partial t} + \frac{\partial\left(\rho u_j u_i\right)}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\eta\left(\dot{\gamma}\right)\frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} \tag{3.7}$$

which comprises (from left to right) the following terms: unsteady, advective, diffusive and pressure source.

3.3 THE NUMERICAL CALCULATION PROCEDURE

The governing equations were discretized following the FVM for unstructured meshes, which starts by integration on each control volume (Ω) and time step (Δt). In this way, the integral form of the equation can be written as a sum of integrals of each term, which is described hereafter.

For the momentum conservation equation (Eq. 3.7) the unsteady term is considered constant inside the control volume, and its integration over the time interval is discretized using a first order (backward) differencing scheme (Versteeg and Malalasekera, 2007). This simple approach was employed since the problems solved are steady and time evolution is used just for relaxation purposes.

To avoid the decoupling between pressure and velocity fields, the normal component of velocity at the cell faces, required for the advective term, is computed using the Rhie and Chow (Versteeg and Malalasekera, 2007) interpolation scheme, whereas scalar velocity is computed using a Total Variation Diminishing (TVD) interpolation scheme (Versteeg and Malalasekera, 2007).

The gradients required for the advective, diffusive and pressure source terms were computed using the least square method and the values from the previous time step (Versteeg and Malalasekera, 2007). For the momentum equation four boundary conditions were considered:

- inlet, with imposed velocity $u_i = u_{i,inlet}$;

- outlet, null normal gradients $\frac{\partial u_i}{\partial x_i} n_i = 0$;

- wall, with null velocity $u_{i,wall} = 0_i$ and

- symmetry, $u_{i,wall} n_{i,wall} = 0$.

As mentioned before, to couple velocity and pressure fields, the normal velocities of the inner faces are computed using a Rhie and Chow (Versteeg and Malalasekera, 2007) based interpolation scheme:

$$\begin{aligned} & u_{j,f} n_{j,f} = \\ & \tfrac{u_{P,f}+u_{nb,f}}{2} n_{j,f} \\ + \quad & \tfrac{1}{2}\left(\tfrac{\Delta V_P}{a_P} + \tfrac{\Delta V_{nb}}{a_{nb}}\right) \tfrac{p_P - p_{nb}}{\|\xi\|} \\ - \quad & \tfrac{1}{2}\left(\tfrac{\Delta V_P}{a_P}\left(\overline{\tfrac{\partial p}{\partial x_j}}\right)_P + \tfrac{\Delta V_{nb}}{a_{nb}}\left(\overline{\tfrac{\partial p}{\partial x_j}}\right)_{nb}\right) \tfrac{\xi_{j,f}}{\|\xi\|} \end{aligned} \tag{3.8}$$

where the gradient $\frac{\partial p}{\partial x_j}$ values are estimated with least squares method, ΔV_P is the P cell volume, ΔV_{nb} is the neighbor's cell volume, a_P and a_{nb} are the diagonal coefficients of cells P and nb of the momentum conservation equation (Versteeg and Malalasekera, 2007).

The mass conservation equation (Eq. 3.1) is used to obtain a pressure correction equation for each control volume. For this purpose, the equation is integrated on the control volume, then, the Gauss' Divergence Theorem is applied and the velocity is written as a sum of an estimated ($\mathbf{u}^*$) and correction ($\mathbf{u}'$) velocities:

$$\sum_f A_f \left(u^*_{j,f} n_{j,f} + u'_{j,f} n_{j,f}\right) = 0 \tag{3.9}$$

Computing the difference between the correct face velocity values, obtained with Eq. 3.8, and those obtained with a similar one to compute the face estimated values $u^*_{j,f} n_{j,f}$, an expression for the normal velocity face correction is obtained:

$$u'_{j,f} n_{j,f} = \frac{1}{2} \left(\frac{\Delta V_P}{a_P} + \frac{\Delta V_{nb}}{a_{nb}} \right) \frac{p'_P - p'_{nb}}{\|\xi\|} \tag{3.10}$$

Substituting Eq. 3.10 in Eq. 3.9 a equation for the pressure field correction is obtained, for each cell:

$$\begin{aligned} & \sum_f \frac{A_f}{2\|\xi\|} \left(\frac{\Delta V_P}{a_P} + \frac{\Delta V_{nb}}{a_{nb}} \right) \left(p'_P - p'_{nb} \right) \\ = \quad & -\sum_f A_f u^*_{j,f} \cdot n_{j,f} \end{aligned} \tag{3.11}$$

where the right hand side corresponds to the mass imbalance of each cell.

The pressure field correction, together with Eq. 3.10, allow to correct the normal face velocities. For the cell velocities a correction expression can be obtained by subtracting the momentum conservation equations for the estimated and correct fields, which gives:

$$\sum_{nb} a_{nb} u'_{i,nb} + a_P u'_{i,P} = -\frac{\partial p'}{\partial x_i} \Delta V_P \tag{3.12}$$

As in the main approach of SIMPLE (Versteeg and Malalasekera, 2007), if the neighbors contributions are neglected, an explicit expression to cell velocity correction is obtained

$$u'_{i,P} = -\frac{\Delta V_P}{a_P} \frac{\partial p'}{\partial x_i} \tag{3.13}$$

where the pressure correction gradient is computed using a least squares method (Versteeg and Malalasekera, 2007).

The full calculation procedure is illustrated in the flowchart depicted in Fig. 3.1.

The process starts with estimated values for velocity and pressure $\left(u^0 \text{ and } p^0 \right)$. Then, the momentum conservation equations are solved. Subsequently, the boundary velocities at the outlet are computed using the appropriate boundary condition,

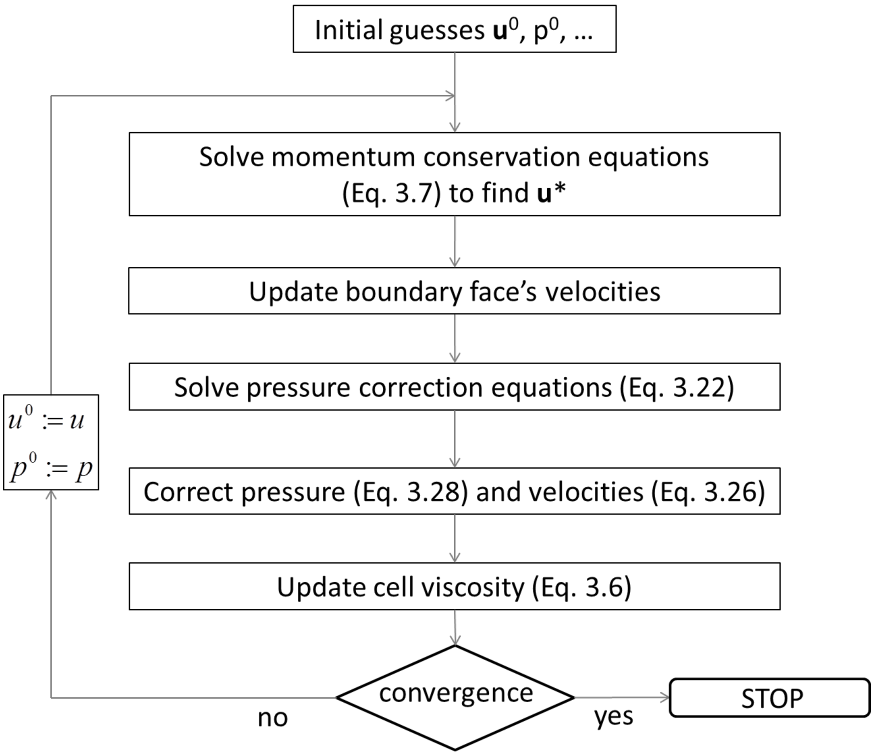

Initial guesses **u**⁰, p⁰, …
Solve momentum conservation equations (Eq. 3.7) to find **u***
Update boundary face's velocities
Solve pressure correction equations (Eq. 3.22)
Correct pressure (Eq. 3.28) and velocities (Eq. 3.26)
Update cell viscosity (Eq. 3.6)
convergence
no
yes
STOP
u⁰ := u
p⁰ := p

Figure 3.1: Code flowchart

and then corrected in order to equalize the outlet flow to the inlet flow. After updating the velocities on boundaries, the pressure correction system is built and solved, enabling to perform the correction of pressures and velocities. To avoid divergence, the pressures on the cells are corrected with a relaxation factor (Versteeg and Malalasekera, 2007) α (it is common to use a value of 0.1), through the following equation:

$$p = p^* + \alpha p' \tag{3.14}$$

The process restarts if the error residuals are larger than a predefined value.

3.4 Code verification

To verify the code, three typical benchmark problems were tested: the Poiseuille flow between parallel plates, which has an analytical solution, and the Lid-driven Cavity and flow around a cylinder problems, whose numerical solutions are given by Ghia et al. (1982) and Bharti et al. (2006), respectively.

3.4.1 *Poiseuille flow between parallel plates*

This problem was solved considering a Generalized Newtonian fluid that follows a Power-law constitutive equation with $n = 1$ (particular case of a Newtonian fluid) and $n = 0.7$. For the mesh sensitivity studies, four meshes were employed with 5, 10, 20 and 40 cells along the channel thickness, which were designated by M0, M1, M2 and M3, respectively .

The numerical values predicted for the axial pressure gradient, p_x, and maximum channel velocity (u_{max}) were compared with their analytical counterparts given by Eq. 3.15 and Eq. 3.16. The results obtained, plotted in Fig. 3.3, allow to confirm that the predicted numerical values tend to the analytical ones as the mesh is refined. It is also worth to notice that with M1 the errors obtained are below 2%.

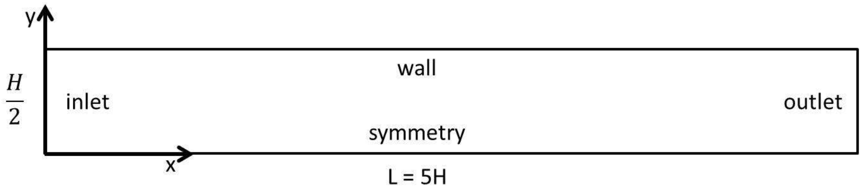


Figure 3.2: Poiseuille problem - geometry and boundary conditions

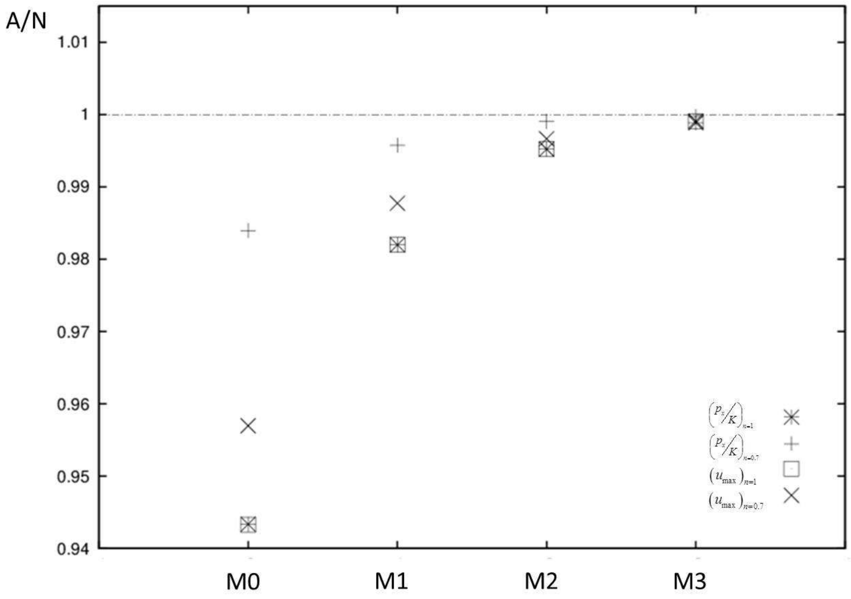


Figure 3.3: Mesh sensitivity study for the ratio between analytical (A) and Numerically predicted (N) values, for maximum velocity u_{max} and axial pressure gradient (p_x/K), for the Poiseuille case study with Newtonian $(n = 1.0)$ and Generalized Newtonain $(n = 0.7)$ fluids

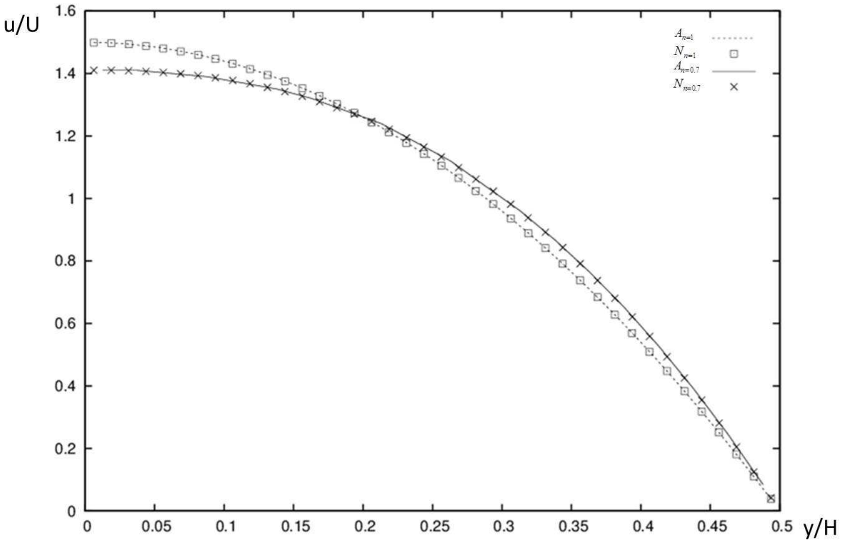


Figure 3.4: Comparison between the Analytical (A) and Numerically predicted (N) normalized velocity profiles, for the Poiseuille case study with Newtonian $(n = 1.0)$ and Generalized Newtonian $(n = 0.7)$ fluids

$$\frac{p_x}{K} = \frac{-\left(\frac{1+2n}{nH}U\right)^n}{H} \tag{3.15}$$

$$u\left(y\right)=\left(\frac{-p_x}{K}\right)^{\frac{1}{n}}\frac{H^{1+\frac{1}{n}}}{1+\frac{1}{n}}\left(1-\left(\frac{y}{H}\right)^{1+\frac{1}{n}}\right) \tag{3.16}$$

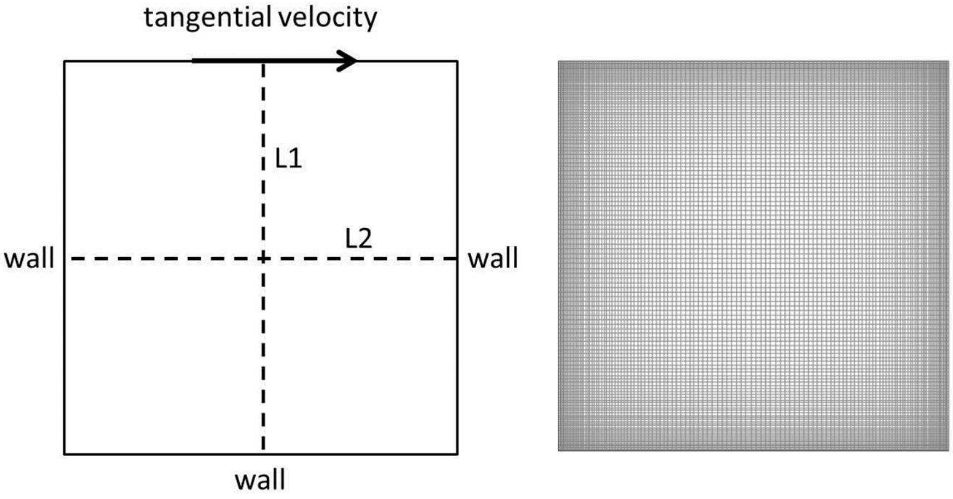
For assessment purposes, the velocity profiles obtained for the two cases considered ($n=1$ and $n=0.7$), with mesh M3, are plotted in Fig. 3.4.

These results allow to confirm the code accuracy when dealing with generalized Newtonian Fluids.

3.4.2 *Lid-driven cavity*

The Lid-driven cavity is another benchmark problem usually employed to verify numerical codes. This problem consists of a square channel with three fixed walls and a tangential homogeneous velocity boundary on the top, as shown in Fig. 3.5a. The flow velocity was defined in order to have a Reynolds number (*Re*) of 100.

A mesh with 129 × 129 cells was used, as illustrated in Fig. 3.5b, with smaller cells close to the walls, where higher velocity gradients are expected.



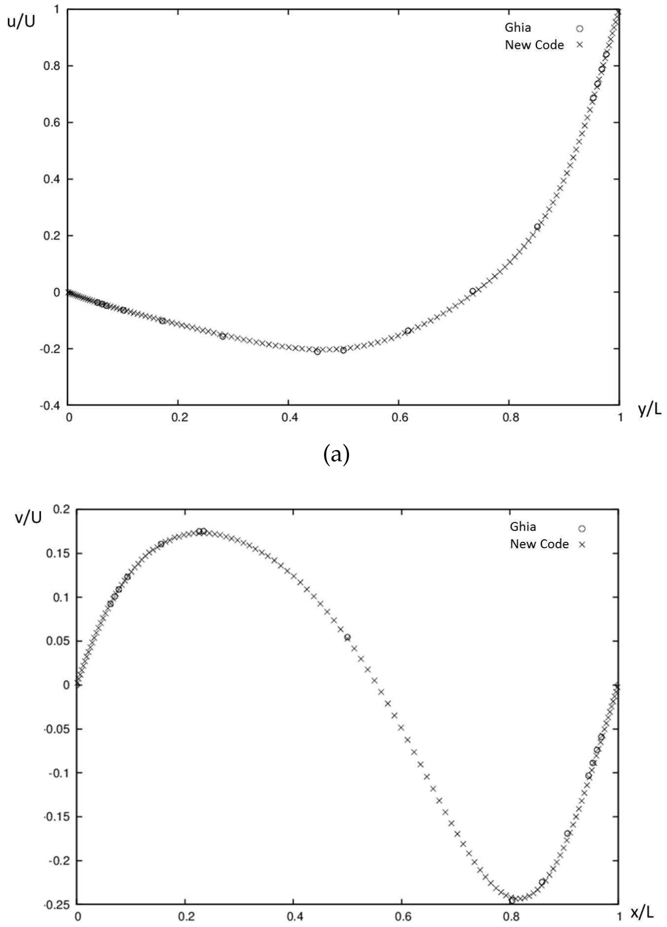
(a)

(b)

Figure 3.5: Lid-driven Cavity: (a) boundary conditions; (b) mesh with 129 × 129 cells

The results obtained with the developed numerical code and those obtained by Ghia et al. (1982) for the u and v velocity components along $L1$ and $L2$ lines (see

Fig. 3.5a), are shown in Fig. 3.6. Both velocity components computed with the new code, evidenced a good correlation with those of Ghia et al. (1982).

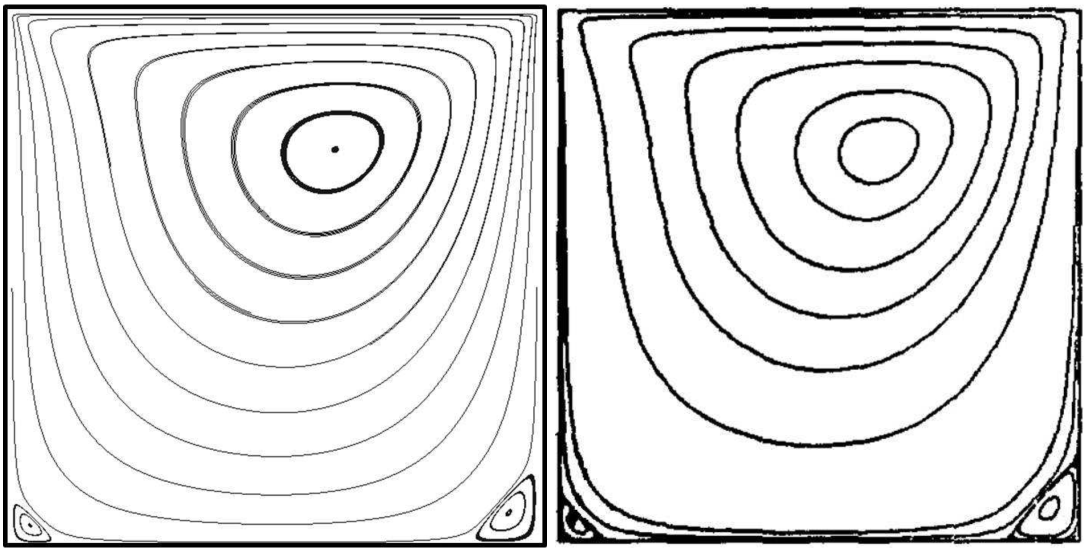


(a)

(b)

Figure 3.6: Velocity for the Lid-driven Cavity case study: (a) u velocity along $L1$ (Fig. 3.5a); (b) v velocity along $L2$ (Fig. 3.5a)

The streamlines predicted by the developed numerical code and by Ghia et al. are presented in Fig. 3.7a and Fig. 3.7b, respectively. The similarity of both results, in terms of number, position and size of the vortices, comprises an additional validation for the implementation.



(a) (b)

Figure 3.7: Lid-driven Cavity problem - streamlines: (a) new code; (b) Ghia et al. (1982)

3.4.3 *Flow around a cylinder*

The flow around a cylinder, one of the most extensively studied problems in fluid mechanics (Panda and Chhabra, 2010; Bharti et al., 2006), was used in this work to verify the accuracy of the numerical predictions on a complex geometry.

The geometry used was, as suggested by Bharti et al. (2006), a square with edge lengths $2L$, having a circle of diameter D in its center, as illustrated in Fig. 3.8. As shown in this figure, due to symmetry reasons just half of the geometry was considered for the numerical model, and adequate boundary conditions were employed. The other boundary conditions set, were a constant velocity of $1\, m \cdot s^{-1}$ for the inlet, null gradients at the outlet and a no slip wall condition, $\mathbf{u}_{wall} = \mathbf{0}$, on the cylinder wall.

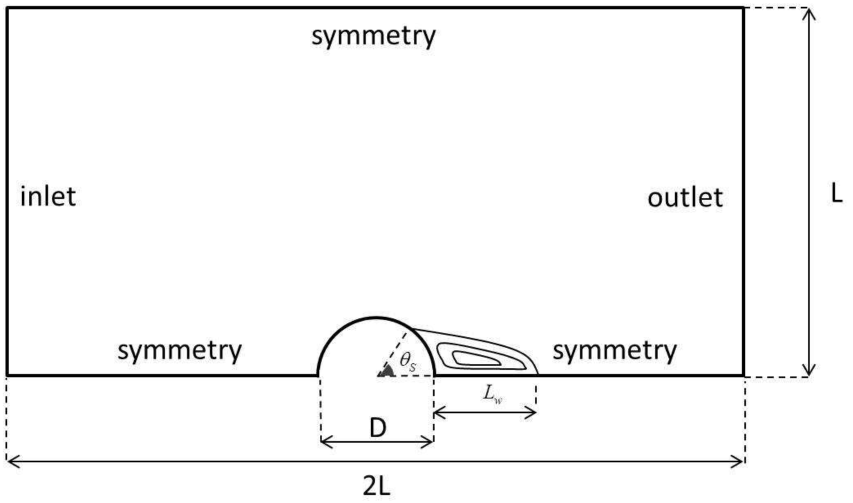


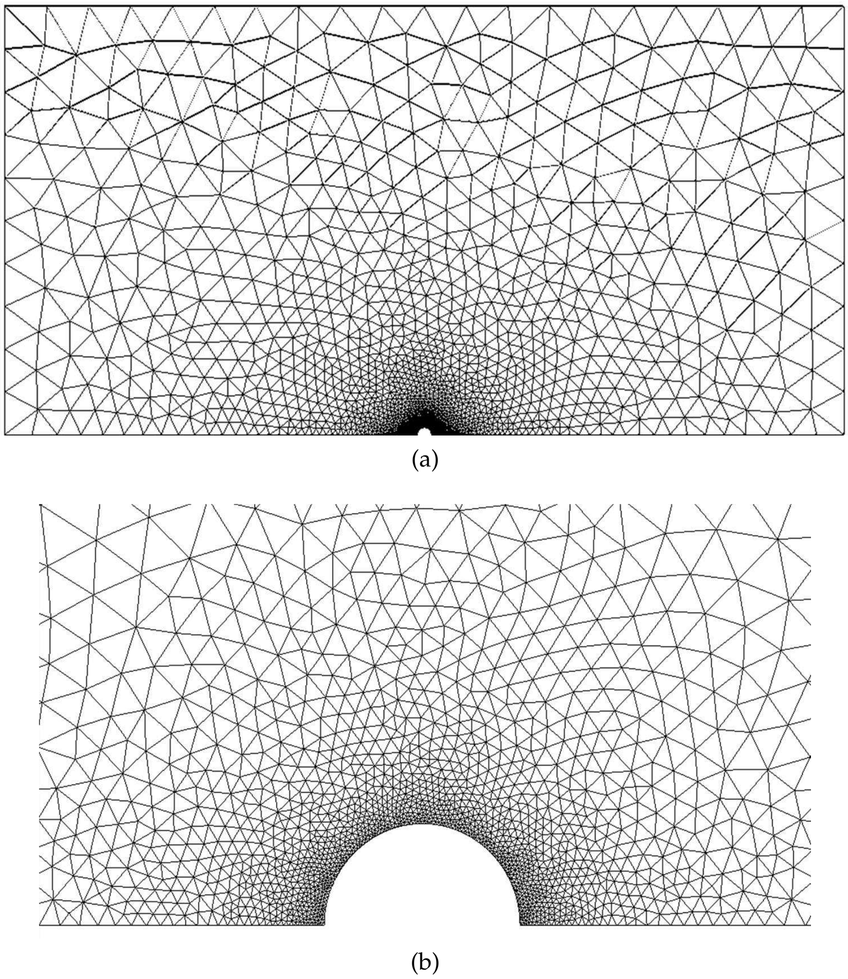
Figure 3.8: Flow around a cylinder problem - geometry and boundary conditions

As suggested by Bharti et al. (2006), a geometry with $L = 30.5D$ was used, since it was found adequate to produce almost domain independent results.

For the numerical models, the coarsest mesh employed (M0) has 75 cells along each symmetry line, 10 cells along each vertical boundary, 20 cells along top boundary (Fig. 3.9a) and 100 cells along the cylinder edge (Fig. 3.9b). On the bottom symmetry lines a progression factor was used.

The meshes M1, M2 and M3 were obtained based on M0, successively doubling the number of cells along each boundary. The number of cells in each mesh is 5148, 21488, 88854 and 355298, respectively, for M0-M3. Here it is important to notice that for the benchmark problem considered, Bharti et al. (2006) used 101 cells along the cylinder edge, for the finest mesh.

To compare the numerical code predictions with similar results obtained by Bharti et al. (2006), two parameters, illustrated in Fig. 3.8, were calculated: the length of wake (or recirculation) zone (L_w), that is the dimensionless distance measured from the rear of the cylinder to the point of the attachment for the near closed streamline on the downstream symmetry edge, and the angle of separation (θ_S), that is measured from the rear stagnation point to the onset of the flow separation from the solid surface.



(a)

(b)

Figure 3.9: Flow around a cylinder problem - mesh: (a) full mesh; (b) close-up view around the cylinder

The results plotted in Fig. 3.10 allow to verify that the numerically predicted and benchmark results computed tend to slightly different values. For the finest meshes used in both works circa 3.2% and 1.7% difference was obtained, respectively, for the length of wake (L_w) and angle of separation (θ_S). However, as mentioned before, the meshes employed in this work were much finer than the ones employed by Bharti et al. (2006), e.g., circa 8 times more cells along cylinder edge were used in our meshes. Regarding the number of cells along the cylinder edge, the meshes employed by Bharti et al. (2006) are equivalent to Mo. Additionally, Bharti et al. (2006) refer that their results were obtained with an accuracy of circa $1-2\%$.

This case study allowed to conclude that the developed numerical code allows to perform accurate computations on complex geometries.

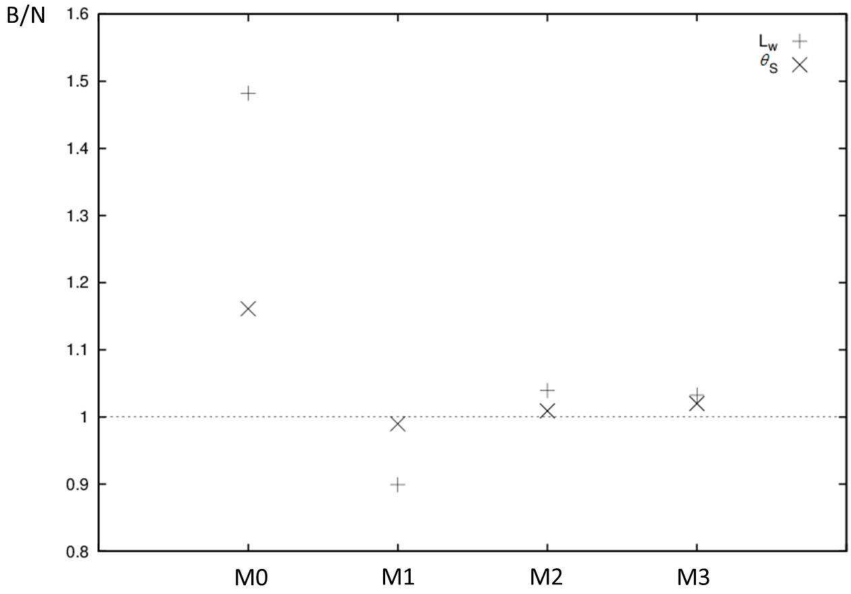


Figure 3.10: Mesh sensitivity study for the ratio between Benchmark (B) and Numerically predicted (N) values, for the length of wake (L_w) and angle of separation (θ_S), for the flow around a cylinder case study

3.5 design of a catheter extrusion die

The developed code was used to optimize the flow distribution in an extrusion die designed for the production of a medical catheter profile, that comprises a complex geometry cross section with five internal channels (lumens), shown in Fig. 3.11a.

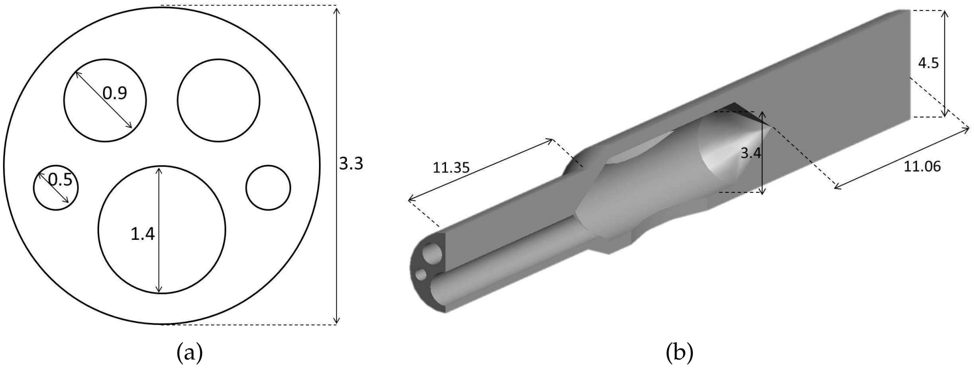


Figure 3.11: Medical Catheter: (a) profile cross section; (b) flow channel geometry (dimensions in mm)

The material employed was a polypropylene homopolymer extrusion grade, Novolen PPH 2150, from Targor. Its rheological behaviour was experimentally characterised in capillary and rotational rheometers, at $230\,^{o}C$(Nóbrega, Carneiro, Pinho and Oliveira, 2003). The shear viscosity data were fitted with the least-squares method by means of the Bird-Carreau constitutive equation (Eq. 3.6), considering $\eta_\infty = 0\,Pa \cdot s$, yielding the following parameters: $\eta_0 = 5.58 \times 10^4\,Pa \cdot s$, $\lambda = 3.21\,s$ and $n = 0.3014$.

The inlet velocity was set in order to have a outlet average velocity of circa $1.2\,m \cdot min^{-1}$.

The full flow channel geometry comprises the region depicted in Fig. 3.11b and an additional convergent circular region that connects this region to the extruder. However, since this convergent region will not influence the flow distribution, the flow was modeled just in the geometry shown in Fig. 3.11b.

Due to the different restrictions promoted to the flow along the channel cross section, which is a direct consequence of the differential thicknesses comprised by the profile, the flow will be naturally unbalanced, resulting in different average linear velocities at the flow channel outlet, which is commonly referred as an unbalanced flow distribution (Nóbrega, Carneiro, Pinho and Oliveira, 2004). Since it is not possible to produce the required geometry when the flow is not properly balanced, one has to find a strategy to obtain a more even flow distribution.

To analyze the flow distribution at the outlet section, the geometry is divided into 4 Elemental Sections (ES) (Carneiro et al., 2004b), as shown in Fig. 3.12a, where the average velocity is computed.

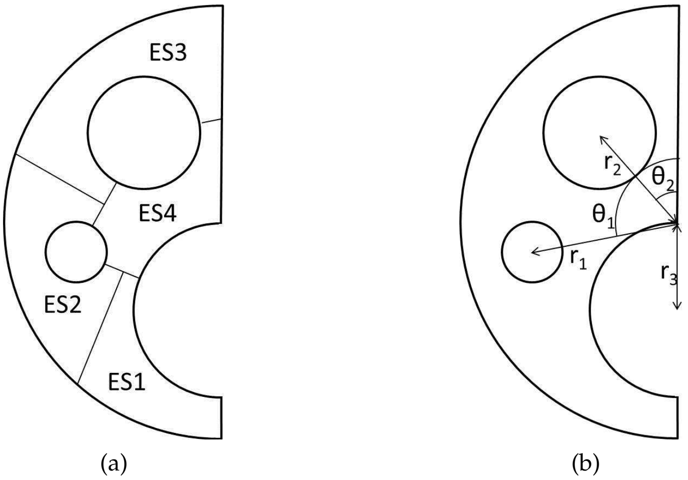


Figure 3.12: Catheter cross section: (a) Elemental Sections; (b) controllable parameters

The quality of the flow distribution was evaluated by an objective function that is always positive and becomes zero when the bulk average velocity is reached in all the ESs that is given by

$$F_{obj} = \sum_i F_{obj,i} = \sum_i \left\| 1 - \frac{\overline{V_i}}{\overline{V}} \right\| \frac{A_i}{A} \tag{3.17}$$

where $\overline{V_i}$ is the average flow velocity at ES_i, $\overline{V}$ is the global outlet average flow velocity, A_i is the area of ES_i and A is the total outlet area.

The contribution of each ES to the objective function can be quantified by $F_{obj,i}$ that gives information on the local flow unbalance. These values were employed to guide the flow channel geometry corrections. Thus, for each trial, the ES that had a greater contribution to the objective function was identified, and the appropriate controllable geometric parameters(s) that affect its velocity was (were) changed, with a view to improve its performance.

The relevant dimensions of the cross-section of the catheter are shown in Fig. 3.11a. In order to be able to balance the flow distribution at the channel outlet, some geometric parameters can be changed to control the flow distribution. These are the angles and distance to the cross-section centre, one for each hole (lumen), as illustrated in Fig. 3.12b. The changes in these parameters are not expected to affect the catheter performance, since the channels (lumens) functionality is maintained when their cross area is kept unchanged. The initial values of these parameters were $r_1 = 1.15\,mm$, $\theta_1 = 45^o$, $r_2 = 0.9\,mm$, $\theta_2 = 100^o$ and $r_3 = 0.7\,mm$.

To compute the flow field a mesh with circa 390000 cells was used, comprising triangular prisms (circa 166000) in constant cross sector zones and tetrahedrons (circa 224000) elsewhere. The typical meshes used on the outlet face and on the global domain, can be seen in Fig. 3.13a and in Fig. 3.13b, respectively.

The contribution of each ES for the objective function is presented in Fig. 3.14, and the evolution of the average velocity in each ES, normalized by the outlet global average velocity, is presented in Fig. 3.15, where it can be seen that after five trials a similar average velocity was attained in all ES. The methodology used to improve the performance of the die in subsequent trials was previously described. According to

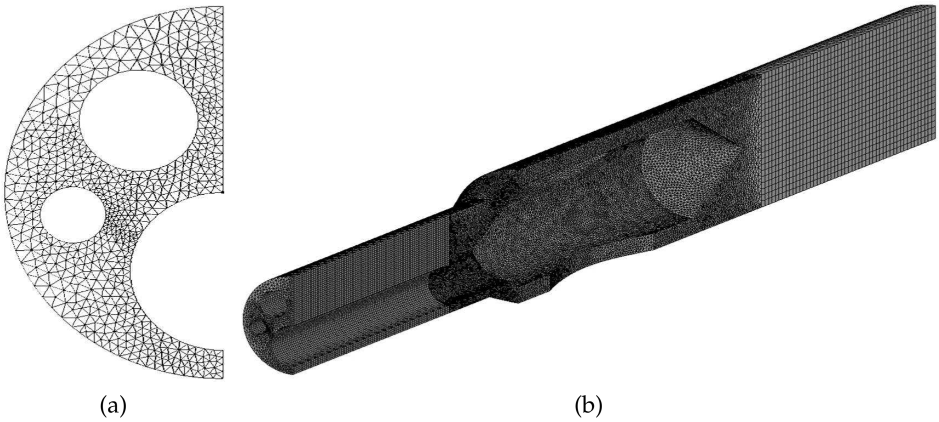

(a) (b)

Figure 3.13: Catheter mesh: (a) outlet face mesh; (b) global domain mesh with circa 390000 cells

it, and as an example, in Fig. 3.14, one can see that on trial T0, ES3 is the one that contributes more to the global objective function, and in Fig. 3.15, it can be seen that it has the highest average flow velocity. To correct this problem, on trial T1 a change on the controllable parameters was done (Fig. 3.16a), in order to increase the restriction of the flow in this ES (ES3). There were several possibilities to achieve this objective, including a change on r_2 (trial T1), θ_2 (trial T2) or a combination of both (trial T3). As one can see in Fig. 3.17, the change on θ_2 led to better results than the change on r_2, and the combination of changes in both parameters resulted even in a better improvement on performance. Having this in mind, and because the ES with higher contribution to objective function remained ES3, on trial T4 θ_2 was changed again. Finally, as the main contribution to the objective function changed to ES4, and its average velocity is higher than those of other ES average velocities, r_3 was decreased on trial T5 in order to restrict the flow in this ES.

As one can see in Fig. 3.17, after five trials the objective function decreased circa one order of magnitude. The improvements obtained for the velocity field are depicted in Fig. 3.16b, evidencing that the final flow distribution is significantly better than the initial one.

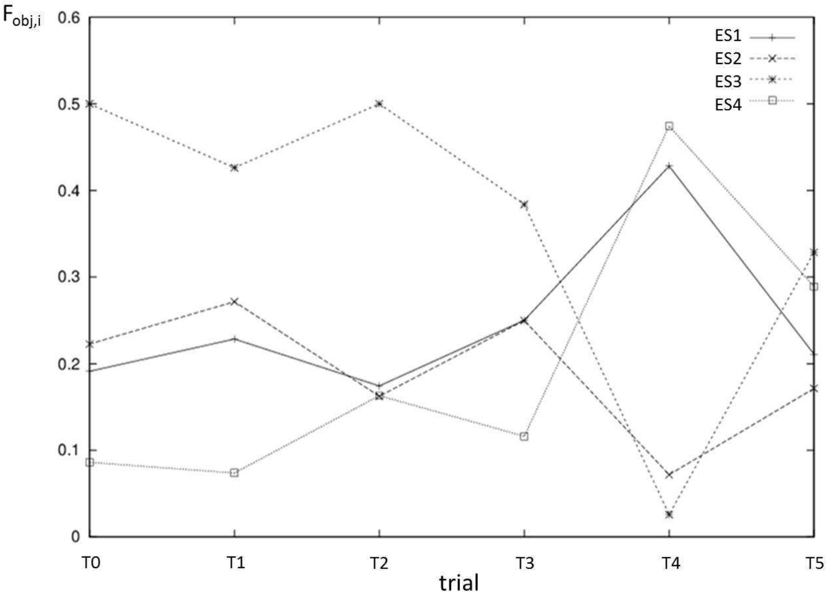
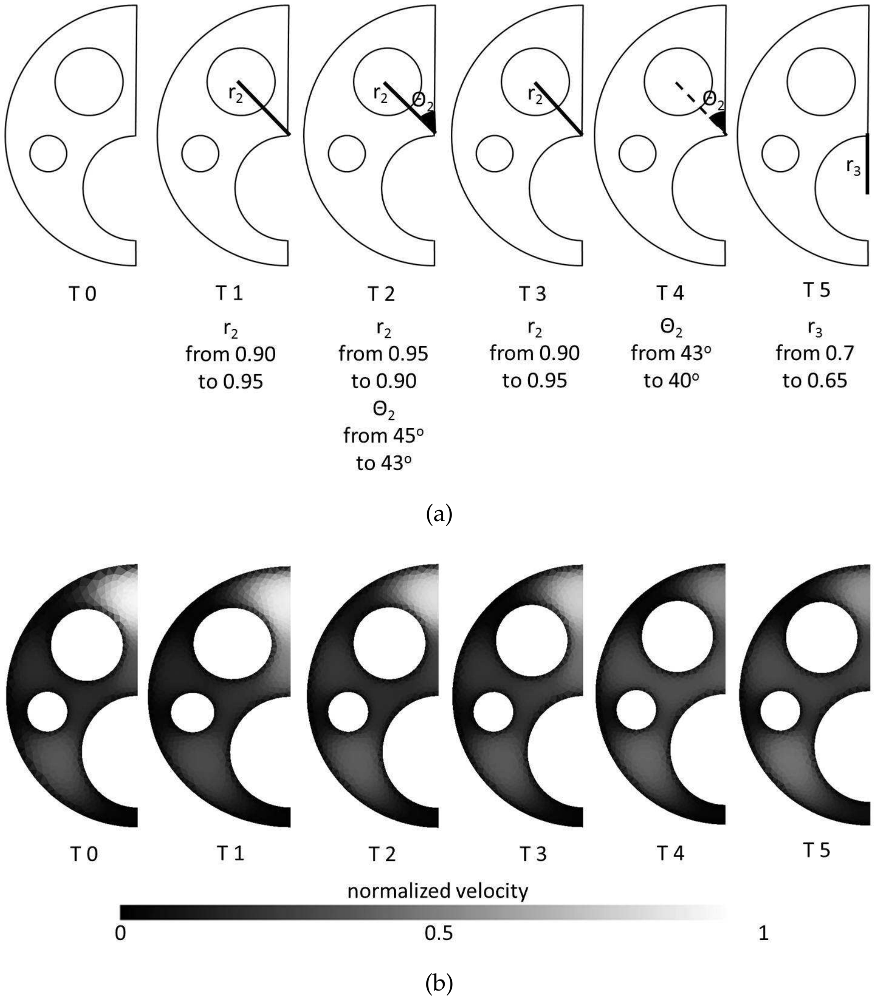


Figure 3.14: Contribution of each ES for the objective function



(a)

(b)

Figure 3.16: Catheter - outlet: (a) geometry changes; (b) outlet velocity field

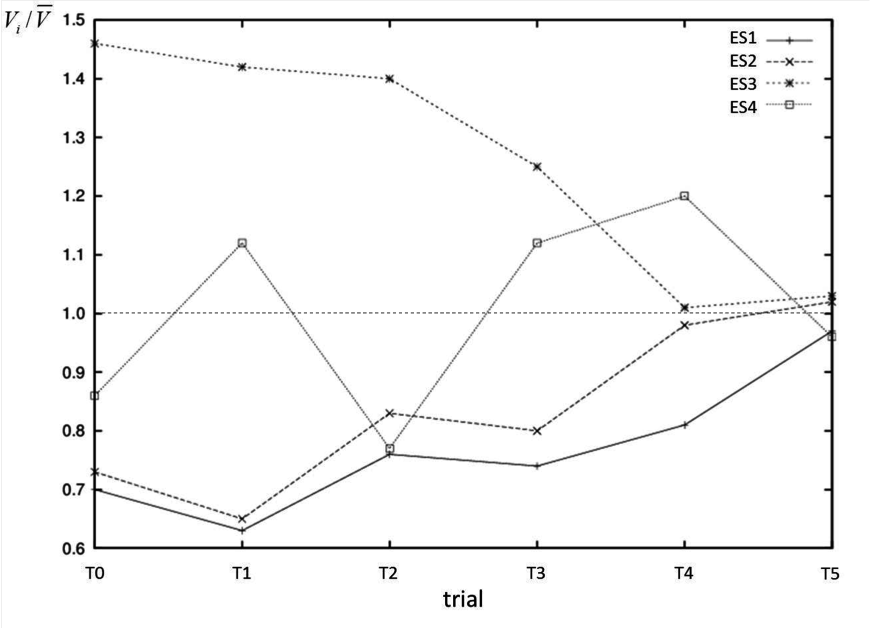


Figure 3.15: Outlet average velocity (normalized by outlet global average velocity)

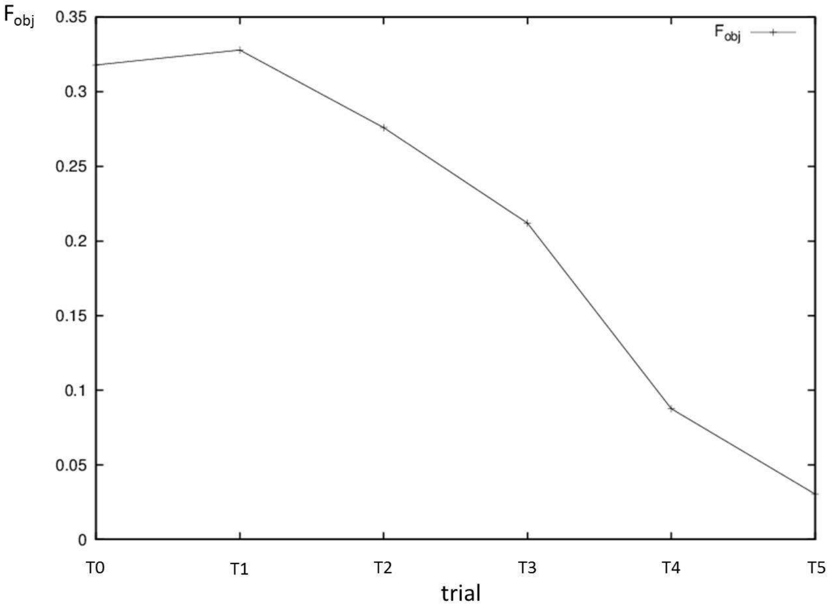


Figure 3.17: Objective function evolution in consecutive trials

3.6 CONCLUSIONS

In this work a numerical code able to deal with complex geometries and unstructured meshes, developed to predict the flow field of polymer melts in confined channels, using generalized Newtonian flow models, was described and verified. The developed code was then used to improve the flow distribution in a complex profile extrusion die, for the production of a medical catheter. The results obtained show that the developed numerical code is a very useful tool to aid the design of extrusion dies used in the production of complex geometry profiles.

4

DESIGN AND OPTIMIZATION OF AN EXTRUSION DIE FOR THE PRODUCTION OF WOOD-PLASTIC COMPOSITE PROFILES

Abstract[1]

In this work the optimization of an extrusion die designed for the production of a wood plastic composite (WPC) decking profile is investigated. The optimization was carried out with the help of numerical tools, more precisely, by solving the continuity and momentum conservation equations that govern such flow, and aiming to balance properly the flow distribution at the extrusion die flow channel outlet. In order to capture the rheological behavior of the material we used a Bird-Carreau model with parameters obtained from a fit to the (shear viscosity versus shear-rate) experimental data, collected from rheological tests. To yield a balanced output flow, several numerical runs were performed by adjusting the flow restriction at different regions of the flow-channel parallel zone cross section. The simulations were compared with the experimental results and an excellent qualitative agreement was obtained, allowing, in this way, to attain a good balancing of the output flow and emphasizing the advantages of using numerical tools to aid the design of profile extrusion dies.

4.1 INTRODUCTION

In the plastic industry, the extrusion process is the most adequate processing technology for the production of constant cross-section products, such as profiles. Due to the emergence of new and demanding applications, profiles with increasing com-

[1] N. D. Gonçalves, P. Teixeira, L. L. Ferrás A. M. Afonso, J. M. Nóbrega, O. S. Carneiro (2013). Design and Optimization of an Extrusion Die for the Production of Wood-Plastic Composite Profiles, submitted to Polymer Engineering and Science.

plexity are being considered for production. These profiles usually comprise complex sections composed of sections with different thickness, which naturally promotes an unbalanced flow. This complexity requires a high expertise on the die design process, in order to diminish the number of trials, often performed experimentally, needed to obtain an even flow distribution. This design optimization stage is thus usually based on trial-and-error procedures that rely on the designer's knowledge and experience, being time and material consuming.

Some of the most relevant studies that can be found in the literature are briefly described hereafter: Hurez et al. (Hurez et al., 1996) developed a design procedure (a blend of the network approach and the cross-section method) for profile extrusion dies. This was based on the resolution of the flow inverse problem, by finding the channel topology that promotes a balanced flow distribution at the die outlet section, from the knowledge of the die contour. They applied the method for the flow balancing of multiple channel dies, which was found to perform very well in all the cases considered. Chen et al. (Chen et al., 1997) used the Taguchi method to investigate the influence of the material, die geometry, and processing conditions on the optimization of a coat-hanger die, using a power law rheological model. Na and Lee (Na and Lee, 1997) performed the shape optimization of a polymer extrusion die by doing simulations with a numerical code based on the finite element method and using a solution strategy of inverse problem for the optimization. The problem of finding the optimum solution was formulated as a quadratic optimization problem. Yu and Liu (Yu and Liu, 1998) developed a simple numerical approach, based on the lubrication approximation, for the optimization of a linearly tapered coat-hanger die. Their approach does not require the analytic pressure drop/flow rate equations being, therefore, suitable for any generalized Newtonian fluid. To verify their results they used 3-D finite element simulations of the complete geometry and a good agreement was found. Ulysse (Ulysse, 1999) used analytical sensitivity and mathematical programming techniques to determine the optimal bearing length to achieve a uniform die exit flow in a 2D two-outlet benchmark problem. The analysis was isothermal and the power-law model was used for the constitutive equation, but later this design model was extended to include temperature distribution calculation and thermal effects were introduced in the analytical sensitivity formulations (Ulysse, 2002).

More recently, Lehnhäuser and Schäfer (Lehnhäuser and Schäfer, 2005) presented a generic optimization technique by considering the steady state flow of an incompressible Newtonian fluid in an arbitrary domain. The integrated shape optimization scheme consisted of three main components: the numerical flow simulator, the mathematical optimization tool, and an approach for efficiently altering the shape of the underlying geometry. Carneiro et al. (Carneiro et al., 2001) presented a quite complete optimization methodology based on numerical methods, that could deal with rheological defects, output flow balance and thermal degradation. Michaeli et al. (Michaeli et al., 2001; Michaeli and Kaul, 2004) proposed a method based on the evolution strategy algorithm, network theory and genetic algorithms. Nóbrega et al. and Carneiro et al. (Nóbrega, Carneiro, Pinho and Oliveira, 2003; Carneiro, Nóbrega, Oliveira and Pinho, 2003; Nóbrega, Carneiro, Pinho and Oliveira, 2004), presented a series of papers where they used simulations obtained with a numerical code, based on the finite volume method, to optimize extrusion die flow channel geometry. Their method is automatic, based on algorithms that iteratively evolute in the steepest gradient direction searching for a minimum/maximum of a predetermined function, and can be undertaken using two alternative design strategies, based either on the die parallel zone length or thickness optimization. Sienz et al. (Sienz et al., 2006) presented a study of a slit die performance, taking into account the coupling of melt flow and die body deflection due to melt pressure. They compared the results obtained with a genetic algorithm, a multipoint approximation optimization technique, and a gradient-based optimization technique in order to find out the best method to reach an optimum melt flow distribution. Lebaal et al. (Lebaal et al., 2009) investigated the optimization of a coat-hanger die by using a computational approach that incorporates three-dimensional finite element simulations and includes an optimization algorithm based on the global response surfaces, with the Kriging interpolation and SQP algorithm. Soury et al. (Soury et al., 2009) presented an optimization method similar to the one used in this work (a numerical based trial-and-error procedure). Their work was oriented for the optimization of an extrusion die for the production of an I-shaped wood-plastic composite profile. They used both numerical methods (finite element method based numerical code) and experimental validation, showing that this combination could yield an acceptable quality profile. Sienz et al. (Sienz et al., 2010) proposed a procedure for implementing sensitive analysis in the design of profile extrusion dies. They showed that the derivatives necessary to evaluate the sensitivity could be obtained through a post-processing step and by the same finite element solver used to solve the discrete governing equations. Mu et al. (Mu et al., 2010) developed an optimization strategy where they used finite element simulations based on the penalty method and the back-propagation neural network. For the search of the global optimal design variables a non-dominated sorting genetic algorithm II (NSGA-II) was used, while the objective functions were evaluated by the established neural network model. Their optimization process was successfully applied to the die design of tubular products using a low-density polyethylene.

Very recently, Liu et al. (Liu et al., 2012) investigated and optimized a multi-cavity aluminum profile for high-speed train, using numerical simulations (Arbitrary Lagrangian Eulerian algorithm), which were experimentally assessed. They performed simulations in order to obtain the optimum profile and, then, with few trial extrusion runs, they were capable to manufacture the required profile. Elgeti et al. (Elgeti et al., 2012) developed a new design approach based on numerical shape optimization, showing a significant reduction in the number of experiments needed to design each new die, using, as an example, a slit profile and a floor skirting. They also compared the results obtained for a Newtonian fluid and a non-Newtonian fluid modelled by the Carreau model.

In this work we optimize the design of an extrusion die for the production of a wood plastic composite (WPC) profile, making use of in-house developed numerical tools, being the results experimentally assessed. For that purpose, we perform the simulations with a finite volume based code, being the subsequent geometry corrections executed as a function of the resulting flow distribution at the die outlet cross-section. To guide the search of the optimum solution, an objective function that quantifies the quality of the die outlet cross-section flow distribution was used.

The remaining of this chapter is structured as follows. In Section 4.2 we briefly describe the numerical code and in Section 4.3 the geometry of the profile, the experimental setup and the rheological characterization of the WPC are described. In Section 4.4 the optimization methodology is briefly explained. In Section 4.5 we discuss the results obtained and the chapter ends, with the main conclusions, in Section 4.6.

4.2 NUMERICAL CODE

The isothermal and incompressible fluid flow considered in this work is governed by the continuity,

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{4.1}$$

and momentum,

$$\rho \left(\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} \right) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\eta \left(\dot{\gamma} \right) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \tag{4.2}$$

equations, where ρ is the fluid density, t is the time, u_i is the i^{th} velocity component and p is the pressure. The viscosity, η, is a function of the shear rate $\left(\dot{\gamma} \equiv \sqrt{2tr\mathbf{D}^2}, \mathbf{D} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right)$, following a Bird-Carreau constitutive equation,

$$\eta \left(\dot{\gamma} \right) = \eta_\infty + \frac{\eta_0 - \eta_\infty}{\left[1 + \left(\lambda \dot{\gamma} \right)^2 \right]^{\frac{1-n}{2}}} \tag{4.3}$$

where η_0 is the zero shear-rate viscosity, η_∞ is the viscosity at very high shear rates, λ is a characteristic time and n is the power-law index.

The governing equations are discretized using a finite volume procedure, using a SIMPLE based procedure, as presented in Chapter 3, and the resultant systems of equations are solved using an iterative based algorithm. Regarding the boundary conditions, we assume the no-slip velocity $(u = 0)$ at the wall, velocity known at the inlet, symmetry at appropriate planes, when required, and null normal gradients at the flow channel outlet.

4.3 GEOMETRY AND EXPERIMENTAL SETUP

Fig. 4.1 shows the target cross section of the profile to be produced, and Fig. 4.2 illustrates the machined extrusion die, from which the WPC profiles were obtained, and a schematic cut view of the same tool. As can be seen in Fig. 4.2 (a), four torpedoes are used to shape the hollow zones of the extrudate. As described later, the numerical studies were performed both for the complete geometry of the die and for a reduced model that comprises just the flow channel parallel zone (the constant cross section region of the flow channel that ends at the outlet cross section). In both cases, due to symmetry reasons, only half of the geometry was considered for the computational model.

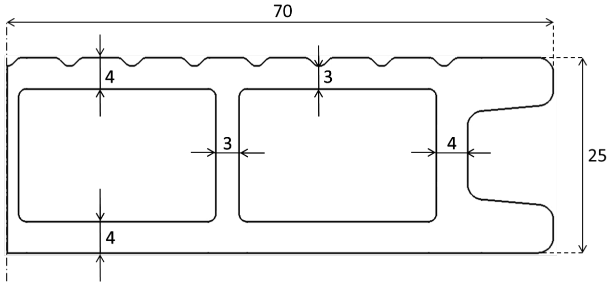


Figure 4.1: Cross section of the profile to be produced with WPC (dimensions in mm). Due to symmetry reasons just half of the flow channel was considered.

A typical mesh used in the simulations comprises a total number of 814032 computational cells, divided into 628508 tetrahedrons and 185524 triangular prisms, for the convergent and constant cross section regions, respectively. As an example, we show in Fig. 4.3(a) the 2D mesh on the parallel zone cross section and, in Fig. 4.3(b) a typical 3D mesh used for the total flow channel.

For the extrusion process a twin-screw counter-rotating intermeshed extruder was used. The extrusion temperature was 190°C and the extrusion flow rate was adjusted in order to obtain an outlet velocity of $0.1\,m\,min^{-1}$. The required dimensions for the

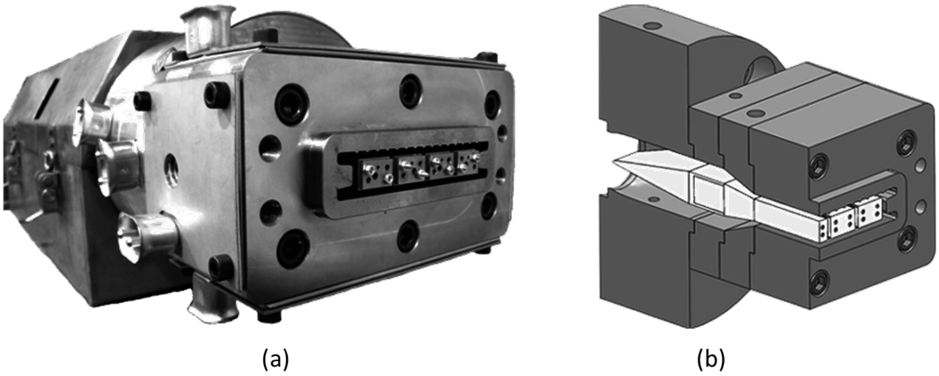


Figure 4.2: Extrusion die used for the production of WPC: (a) actual production tool and (b) cut view of the tool geometry.

profile are given in Fig. 4.1 and it was produced with a WPC formulation comprising 50% of wood particles in a PVC (K57) matrix. The characterization of the composite melt was performed by capillary rheometry. To obtain the Bird-Carreau model parameters we performed a curve fit to the experimental data (shear viscosity versus shear-rate) and the following parameters were obtained: $n = 0.32$, $\eta_0 = 53993\,Pa.s$, $\eta_\infty = 0\,Pa.s$ and $\lambda = 2.36\,s$. The density of the WPC was also characterized and a value of $1200\,kg\,m^{-3}$ was obtained.

For the present study, we first conceived and constructed an extrusion die based on our previous experience in the field, and after, the flow channel geometry was optimized with the support of the flow modelling code. This was done for two reasons: i) two experimental case studies allowed a more extensive experimental validation of the numerical code; ii) the production of profiles with two extrusion dies (non-optimized and optimized) allowed a better evaluation of the benefits obtained when a properly designed extrusion die is employed. It is important to notice that, due to the modular construction adopted for the extrusion die, a new design trial will just require the production of new torpedoes (see Fig. 4.4), thus with a minimum impact on the production cost.

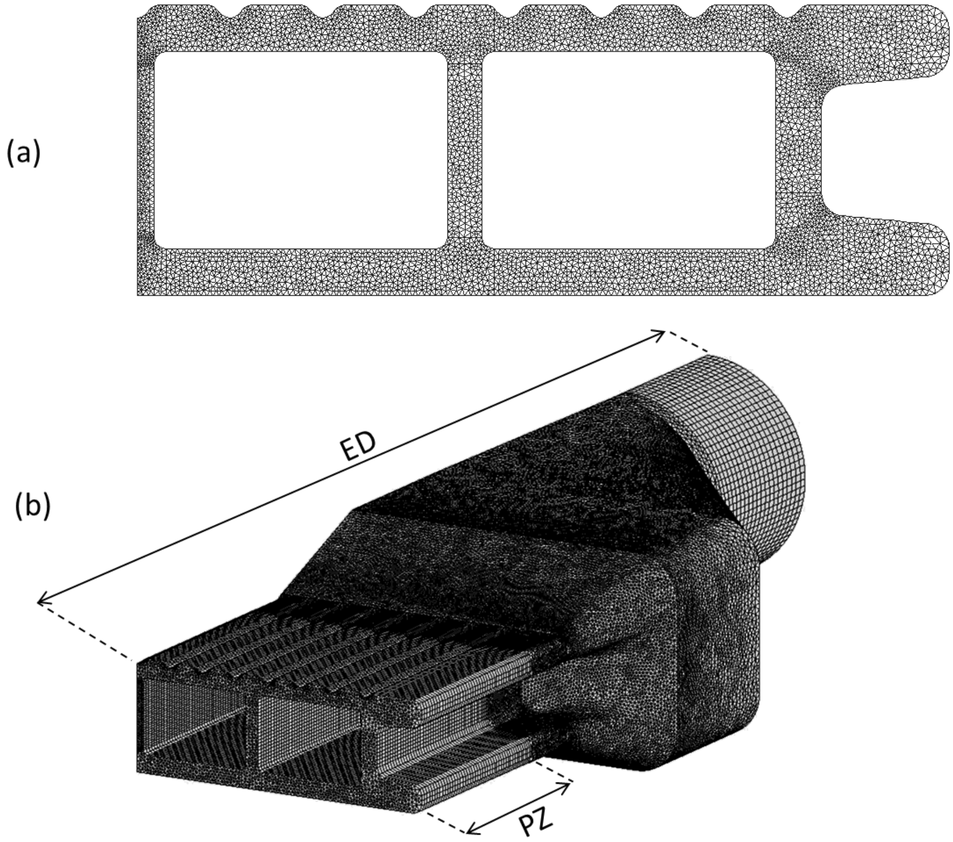


Figure 4.3: Mesh used in the simulations: (a) front view of the parallel zone (PZ); (b) typical 3D view of the flow channel (ED). Due to symmetry reasons just half of the flow channel was modelled.

4.4 optimization methodology

For optimization purposes, the geometry of the parallel zone was divided into 15 elemental sections (ES), as shown in Fig. 4.5, where the local average velocity was monitored.

The objective of the optimization process was to obtain, in each ES, the flow rate that allowed the achievement of the required profile dimensions, as proposed by Nóbrega et al. (Nóbrega, Carneiro, Pinho and Oliveira, 2003). This type of approach, the so called thickness optimization strategy, generates extrusion dies less sensitive to variations of the processing conditions (Carneiro, Nóbrega, Oliveira and Pinho, 2003). For that purpose, modifications of the geometry of the die parallel zone were done by changing the dimensions of the torpedoes cross-section (Fig. 4.5), and, therefore,

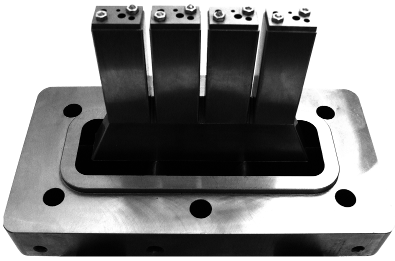


Figure 4.4: Geometry of the torpedoes.

the external contour of the flow channel was not modified. As mentioned before, this procedure allows to reduce the manufacturing costs of each trial die.

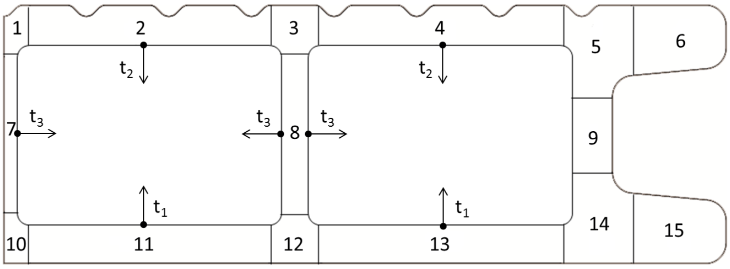


Figure 4.5: Division of the parallel zone geometry into elemental sections (ES), and geometry parameters.

In order to guide the modifications to be performed on the geometry for improving the flow distribution, we devised an objective function, defined as

$$F_{obj} = \sum_{i=1}^{n_{ES}} F_{obj,i} = \sum_{i=1}^{n_{ES}} \left(1 - \frac{Q_i}{Q_{obj,i}}\right)^2 \frac{A_{obj,i}}{A}, \tag{4.4}$$

where n_{ES} is the number of ES, Q_i is the volumetric flow rate, that was obtained from the actual simulation in ES_i, while $Q_{obj,i}$ is the objective flow rate for the ES_i, $A_{obj,i}$ is the required area for the ES_i and A represents the total cross section area required for the profile. Defined in this way, the ratio $A_{obj,i}/A$ is, therefore, the weight of the contribution of ES_i to the global F_{obj}. This F_{obj} is always positive and tend to zero as the flow distribution is improved. It would be zero if in all ES the local flow rate achieved (Q_i) equals the required value ($Q_{obj,i}$). Consequently, the computation of the contribution of each ES to the total objective function, given by $F_{obj,i}$, for a particular trial geometry, allows to identify the ESs that most contribute to the F_{obj} and thus the ones that should be modified on the subsequent trial.

In order to define the design procedure we compared the flow rate distribution obtained when the complete extrusion die flow channel (ED) was modeled, with that corresponding to a model that was limited to the parallel zone (PZ). The results plotted in Fig. 4.6 show that only small differences were found. This happens because the length of the die parallel zone is enough for the achievement of an almost fully developed flow. This fact, together with the demanding computational times required for the ED model (more than $9h$) led to perform the study with the PZ model, which required circa $3h35min$ for the computation.

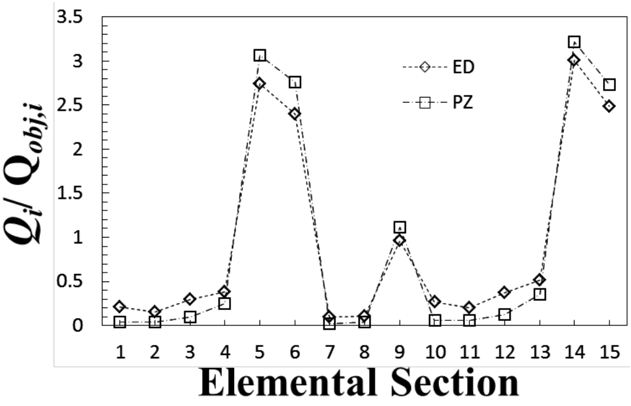


Figure 4.6: Ratio between the computed (Q_i) and required ($Q_{obj,i}$) flow rates, obtained with two computational models: full (ED) and parallel zone (PZ) flow channels.

The iterative procedure adopted in the design methodology can be described as follows:

1. The results obtained with the numerical modeling code, allow to compute the value of $F_{obj,i}$ for each ES, and the global F_{obj}. Then the *ESs* that have the highest values for the ratio $\frac{F_{obj,i}}{F_{obj}}$ are selected for modification. By doing so, we are selecting the *ESs* that have the highest contributions to the deviation from the desired flow distribution. Since the design methodology adopted is iterative, there is no need to perform modifications in all ES at the same time.

2. For each ES selected the sign of the modifications to be performed on local thickness is identified. Thus, when the local flow rate (Q_i) is lower than the required ($Q_{obj,i}$) the thickness is increased, and vice-versa.

3. The third step is to correct the geometry. The adjustable geometry parameters employed are t_i, with $i = 1$ to 3 (see Fig. 4.5), thus each trial is defined by 3 variables. Notice that the dimensions of ES_9 were kept constant on purpose, because this would affect the flow on ES_6 and ES_{15}, which, as will be shown later, are the most problematic ones. The modifications performed in some ES, e. g., ES_2 and ES_4, were forced to be equal in order to facilitate the machining of the torpedoes.

The process is repeated until an acceptable balanced flow distribution is achieved.

4.5 RESULTS AND DISCUSSION

The outlet dimensions of the first trial die (Fig. 4.7 (trial 1)) flow channel were those required for the profile (Fig. 4.1).

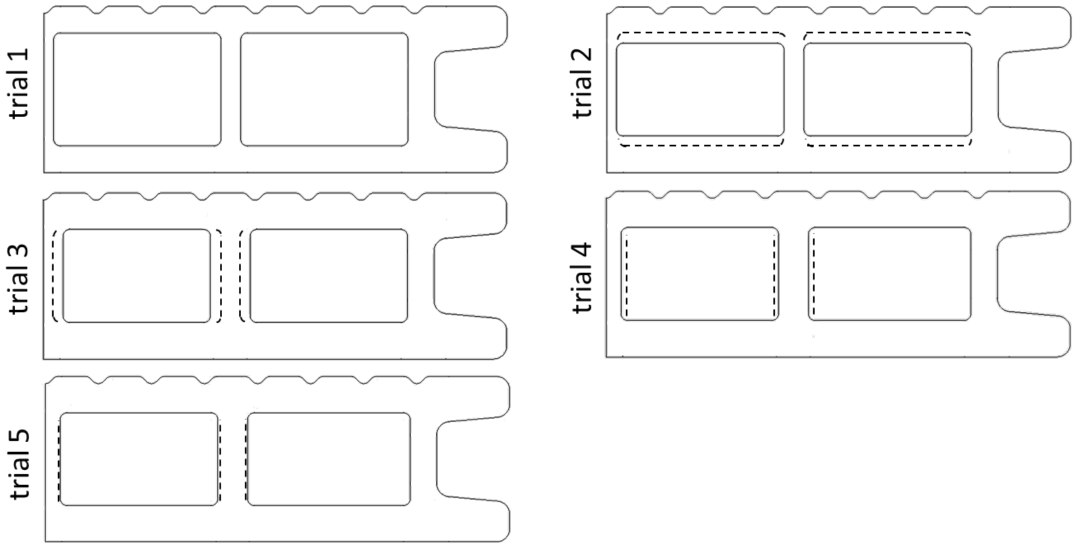


Figure 4.7: Geometry modification of the die parallel zone cross section, along the optimization process (the previous trial geometry appears as a dashed line).

The numerical results obtained with this initial geometry (Fig. 4.8 (trial 1)) show that the distribution of flow at outlet is very unbalanced, with higher velocities in sections where the restrictions to the flow are lower.

A similar information can be obtained in Fig. 4.9 (b), where it can be seen that the ratios $Q_i/Q_{obj,i}$ of the ES_5, ES_6, ES_{14} and ES_{15} are circa 3, indicating that the flow rate

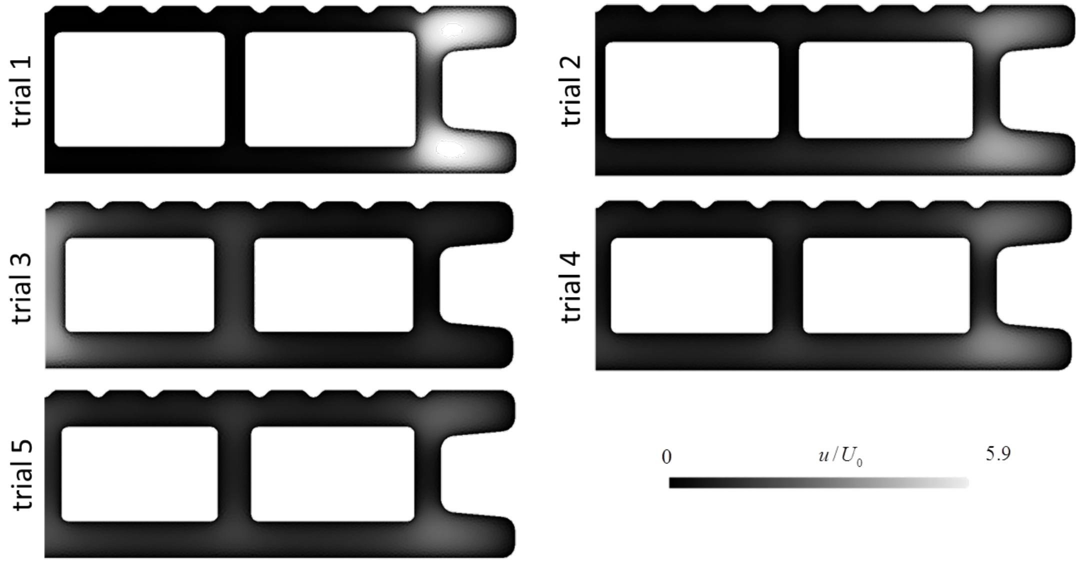


Figure 4.8: Evolution of the velocity distribution along the optimization process at the flow channel outlet.

in those ES are circa 3 times higher than the required. Analyzing the ratios $F_{obj,i}/F_{obj}$ in Fig. 4.9 (a), it can be seen that the ES that most contribute to the high value of the objective function are the previously mentioned, ES_5, ES_6, ES_{14} and ES_{15}, indicating that the geometry changes that affect the flow in these ESs are expected to result on the largest reduction of the F_{obj} value.

In order to reduce the flow rate in the most problematic ESs, the dimensions of the remaining ESs were increased (Fig. 4.5), by setting $t_1 = t_2 = 1.5\,mm$ (Fig. 4.7 (trial 2)). With these modifications, the flow ratio (Q_i) in ES_5, ES_6, ES_{14} and ES_{15} approached their respective objective value ($Q_{obj,i}$), as can be seen in Fig. 4.9 (b).

In trial 2, the new group of elemental sections with higher values of $F_{obj,i}/F_{obj}$ is $\{2, 5, 8, 14\}$ (see Fig. 4.9 (a)). In this trial the flow rate is larger than 1 in ES_5 and ES_{14} and smaller than 1 is ES_2 and ES_8 (see Fig. 4.9 (b)). Thus, for the subsequent geometry, an increase of ES_7 and ES_8 thicknesses was tested, by setting $t_3 = 1.5\,mm$, as illustrated in Fig. 4.7 (trial 3).

The new simulation (trial 3) revealed a flow distribution worst than the previous one (trial 2), promoting an increase of F_{obj} (Fig. 4.10), with high flow rates at the profile center region (see Fig. 4.8 (trial 3) and Fig. 4.9 (b)), more specifically in ES $\{1, 7, 10\}$. Since the ratio $Q_{obj,i}/Q_{obj}$ is larger than 1 for those 3 ESs, the previous modifications

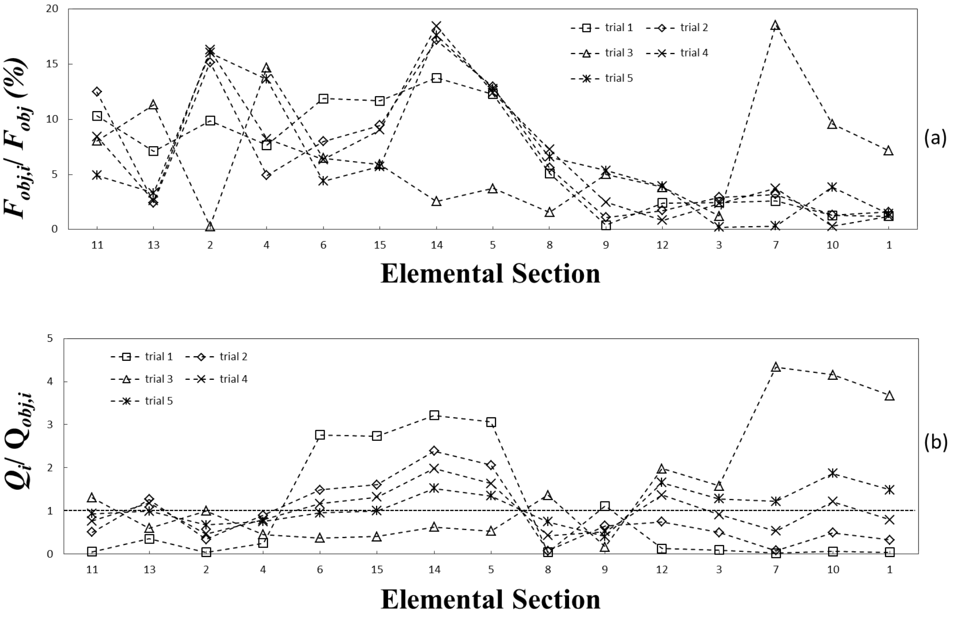


Figure 4.9: Plots of $\frac{F_{obj_i}}{F_{obj}}$ and $\frac{Q_i}{Q_{obj\,i}}$ for the different ES along the five trials. The *ESs* were ordered by descending area.

were redefined by setting $t_3 = 0.75\,mm$ (Fig. 4.7 (trial 4)). This correction resulted in an additional reduction of the objective function $\left(F_{obj}\right)$ (Fig. 4.10), but the flow distribution stills slightly unbalanced (Fig. 4.8 (trial 4) and Fig. 4.9 (a) and (b)).

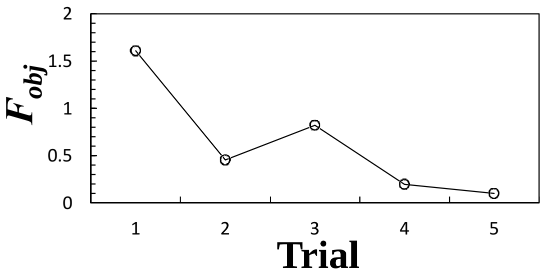


Figure 4.10: Evolution of the global objective function $\left(F_{obj}\right)$ along the optimization process.

From the analysis of Fig. 4.9 (a), the highest contributions to the F_{obj} come from ESs {2, 5, 14}, with a flow rate lower than the required in ES_2 and higher in the two remaining ones. The trial solution to solve these problems was to increase slightly the value of t_3 ($1\,mm$) (Fig. 4.7 (trial 5)). For this new trial the flow distribution (Fig. 4.8, trial 5), flow rate ratios $Q_i/Q_{obj,i}$ (Fig. 4.9 (b)) and F_{obj} (Fig. 4.10) were additionally improved.

The evolution obtained from trial 1 to trial 5, in terms of both the velocity contours (Fig. 4.8) and Fobj (Fig. 4.10), allows to realize the improvements obtained along the design process.

For assessment purposes, the torpedoes required to build the tools corresponding to trials 1 and 5 were machined and tested experimentally. Fig. 4.11 illustrate the profiles obtained with both tools when used in real extrusion runs.

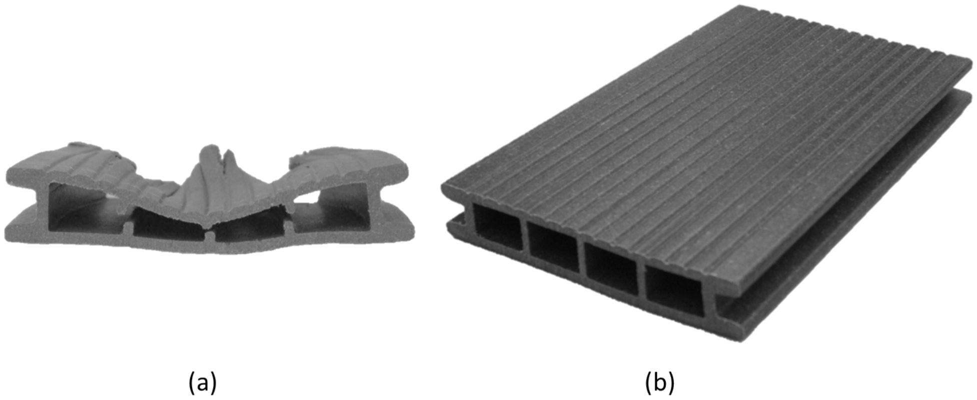


Figure 4.11: Profiles produced during the experimental runs: (a) initial trial geometry (trial 1); (b) optimized geometry (trial 5).

As predicted by the numerical code, the flow distribution obtained with trial 1 die was clear unbalanced, not even allowing to produce a profile in a steady manner. The flow rate obtained in ES_7 and ES_8 was so small, that those regions were not completely filled with material. A similar situation occurred for ES_1 to ES_4 (Fig. 4.1), that in the photo (Fig. 4.11 (a)) seem to be formed, but were not produced continuously, being interrupted several times as an outcome of the low flow rate predicted (and obtained) in these regions. A similar effect was identified for ES_{10} to ES_{13} (Fig. 4.1), but less severe, since this region was continuously extruded, but with a lower thickness than the required. The improvements obtained with the optimized die flow channel were remarkable (see Fig. 4.11 (b)), enabling to produce a continuous profile with all the dimensions close to the required values.

4.6 CONCLUSIONS

In this work, a profile extrusion die for the production of a wood plastic composite profile was designed, being this process carried out by the employment of a numerical modelling code based on the Finite Volume Method, able to cope with unstructured meshes and Generalized Newtonian Fluids, and guided by an objective function. A flow channel geometry, whose parallel zone cross section geometry was similar to that of the desired profile, was taken as a first trial. For optimization purposes, the information computed by the numerical modelling code, namely the contribution of each elemental section for the global objective function, allowed to identify the most critical regions, which were adjusted on the subsequent trial, in order to improve the tool performance. This process showed to be effective and thus should be explored on future implementation of automatic design schemes.

The numerical results obtained for the initial and final die geometries were compared with experimental data, collected in extrusion runs performed with the equivalent tools. A good qualitative agreement between the numerical and experimental results, in terms of the flow distribution, was observed.

The methodology adopted allowed to produce the wood plastic composite profile with the required dimensions.

5

USING THE GPU TO DESIGN COMPLEX PROFILE EXTRUSION DIES

Abstract[1]

In the present work the benefits of using graphics processing units (GPU) to aid the design of complex geometry profile extrusion dies, are studied. For that purpose, a 3D finite volume based code that employs unstructured meshes to solve and couple the continuity, momentum and energy conservation equations governing the fluid flow, together with a constitutive equation, was used. To evaluate the possibility of reducing the calculation time spent on the numerical calculations, the numerical code was parallelized in the GPU. For verification purposes, simulations were performed for three benchmark problems: Poiseuille flow, lid-driven cavity flow and flow around a cylinder. Subsequently, the code was used on the design of two real life extrusion dies for the production of a medical catheter and a wood plastic composite decking profile. To evaluate the benefits, the results obtained with the GPU parallelized code were compared, in terms of speedup, with a serial implementation of the same code, that traditionally runs on the central processing unit (CPU).

5.1 INTRODUCTION

The need for multitask optimization has been, for long, a priority during our evolution. The idea that two persons usually perform better than one, is intrinsic to our existence, and this reflects in our actions, creations and way of life. With the rapid growth of science, sophisticated machines able to perform Men's work were invented, and this

[1] N. D. Gonçalves, S. P. Pereira, L. L. Ferrás, J. M. Nóbrega, O. S. Carneiro (2013). Using the GPU to Design Complex Profile Extrusion Dies, submitted to Computer Methods in Applied Mechanics and Engineering

influenced, in part, our concept of parallel work, extending the parallel multitask concept to the world of computers.

The desire to get more computing power and better reliability by orchestrating a number of low cost computers has given rise to the creation of computational clusters, with this invention being (arguably) attributed to Gene Amdahl of IBM (Amdahl, 1967), who in 1967 published a seminal paper on parallel processing (Amdahl's Law). Until now, the cluster concept continues to hold, but the evolution led to the creation of more powerful computers with more than one core, allowing the employment of the parallel computing concept in a single computer (Reilly, 2003).

When thinking about the optimization of engineering/physics problems, we realize that most of them result in the numerical solution of differential or partial differential equations. This numerical solution is usually expensive because it requires the resolution of large systems of equations. The fact that the physical models are usually nonlinear, forces an iterative procedure, enhancing the need for more computational power.

The current industrial problems are becoming more and more complex on daily basis, consuming computer resources and demanding heavy computations. If we want to obtain acceptable computational times, we must take advantage from all the parallel computational power available in a computer. In this framework, graphics processing units (GPUs), for a long time only seen as powerful tools to enhance the video games graphics, are now a speedup enhancer for large dimension engineering problems (Elsen et al., 2008).

Therefore, since 1999 we have witnessed an increasing interest on GPUs, and the graphics processors have evolved from fixed function pipelines towards fully programmable floating point pipelines (Owens et al., 2008). NVIDIA (NVIDIA, 2013) has developed the CUDA programming toolkit, which includes an extension of the C language and facilitates the programming of GPUs for general purpose applications, by preventing the programmer to deal with the graphic details of the GPU (Castro et al., 2011).

The literature is rich in methods for the GPU parallel computation of a matrix solution. Bolz et al. (Bolz et al., 2003) implemented two basic computational kernels: a sparse matrix conjugate gradient solver and a regular-grid multigrid solver, showing that real-time applications, ranging from mesh smoothing and parametrization to fluid and solid mechanics solvers, could greatly benefit from these. Later, Krüger and Westermann (Krüger and Westermann, 2003) introduced a framework for the implementation of linear algebra operators on programmable graphics processors (GPUs). They proposed a stream model for arithmetic operations on vectors and matrices for the efficient communication on modern GPUs. In order to assess their model, they performed simulations of the 2D wave equation and the incompressible Navier-Stokes equations, using direct solvers for sparse matrices. These two articles are perhaps the most cited on the GPU specific literature. However, there are other works that deserve our attention. Fatahelian et al. (Fatahalian et al., 2004) performed an in-depth analysis of dense matrix-matrix multiplication, which reuses each element of input matrices $O(n)$ times. Although its regular data access pattern and highly parallel computational requirements suggested an efficient evaluation on GPU of matrix-matrix multiplications, they found that these are less efficient than current cache-aware CPU approaches. Hall et al. (Hall et al., 2003) studied more efficient algorithms that make the implementation of large matrix multiplication on upcoming GPU architectures more competitive, using only 25% of the memory bandwidth and instructions of previous GPU algorithms. Ohshima et al. (Ohshima et al., 2007) proposed a new parallel processing environment for matrix multiplications by using both CPUs and GPUs. They decreased 40.1% the execution time of matrix multiplications when compared with using the fastest of either CPU only case or GPU only case. Monakov et al. (Monakov et al., 2010) presented a new storage format for sparse matrices that better employs locality, has low memory footprint and enables automatic specialization for various matrices and future devices via parameter tuning.

A quick literature survey, shows that the use of GPUs for increasing the performance of computations depends on the class of problems we study. In this work we are interested in the numerical solution of the Navier-Stokes equations. Although some limitations exist in regard to Computational Fluid Dynamics (CFD), we can find in the literature successful works regarding the solution of Euler and Navier-Stokes equations.

The first generation of GPU hardware allowed high speedups, but only single precision was used (Hagen et al., 2006; Elsen et al., 2008; Brandvik and Pullan, 2008). For the second generation of GPU, initially lower speedups were reported in the literature because of the employment of double precision numbers (Cohen and Molemaker, 2009; Corrigan et al., 2009), but Kampolis et al. (Kampolis et al., 2010) and Asouti et al. (Asouti et al., 2010) reported double precision speedups for 2D and 3D Navier-Stokes solvers of circa 20x. A very recent paper on the GPU performance for a finite-difference compressible Navier–Stokes solver, suitable for direct numerical simulation (DNS) of turbulent flows, also revealed speedups of 22x. However, in order to obtain such performance, all the above mentioned implementations required a complex and efficient manipulation of the several memories available on the GPU. Aiming to evaluate the performance of the Fermi GPU generation, Pereira et al. (Pereira et al., 2013) obtained maximum speedups of 20x, solving the 2D Navier-Stokes equations together with a inelastic constitutive equation for simple benchmark flows. The results showed that it was possible to obtain a significant better performance, without complex memory manipulations, which are only accessible to hi-skilled programmers.

The objective of this work is to assess the performance of a GPU parallelized 3D Navier-Stokes solver, using inelastic fluids governed by the Bird-Carreau constitutive equation, and its employment on the design of industrially relevant extrusion dies for the production of complex profiles, such as a catheter, for medical applications, and a wood plastic composite profile (WPC), with application on the building industry. In this way, we extend the previous work by Pereira et al. (Pereira et al., 2013) where only simple geometries were tackled, but still using the same straightforward implementation (without any complex memory optimization).

The remaining of this chapter is organized as follows. In Section 5.2 we describe the relevant governing equations, the numerical procedure and the code parallelization. In Section 5.3 we present the code verification, discuss the performance analysis for three benchmark problems and, finally, evaluate the benefits of employing the GPU parallelized code on the design of two extrusion dies for the production of complex profiles. The Chapter ends with the conclusions in Section 5.4.

5.2 GOVERNING EQUATIONS AND NUMERICAL PROCEDURE

5.2.1 *Code implementation*

In this work we consider the isothermal incompressible fluid flow that is governed by the continuity,

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{5.1}$$

and momentum,

$$\rho\left(\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\eta\left(\dot{\gamma}\right)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{5.2}$$

conservation equations, where ρ is the fluid density, t is the time, u_i is the i^{th} velocity component, p is the pressure and $\eta\left(\dot{\gamma}\right)$ is a non-constant viscosity modeled using the Bird-Carreau constitutive equation, given by

$$\eta\left(\dot{\gamma}\right) = \eta_\infty + \frac{\eta_0 - \eta_\infty}{\left(1 + \left(\lambda\dot{\gamma}\right)^2\right)^{\frac{1-n}{2}}} \tag{5.3}$$

where $\dot{\gamma}$ is the shear rate $\left(\dot{\gamma} \equiv \sqrt{2tr\mathbf{D}^2},\ \mathbf{D} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right)$, η_0 is the zero shear-rate viscosity, η_∞ is the viscosity at very high shear rates, λ is a characteristic time and n is the power-law index.

For the wall velocity boundary condition, we assumed the usual no-slip condition,

$$u_i = 0. \tag{5.4}$$

A fully-implicit Finite Volume based numerical method is used to solve Eqs. 5.1 to 5.3. The method employs a time marching pressure-correction algorithm, formulated with a collocated variable arrangement and unstructured meshes (Chapter 3). The governing equations are integrated, in space, over the control volumes (cells with volume V_P) forming the computational mesh, and along time, over a time step (Δt). The volume integration benefits from Gauss Divergence Theorem, and the subsequent surface integrals are discretized, with the help of the midpoint rule, so that sets of linearized algebraic equations are obtained, for each velocity component u_i, having the following general form:

$$a_{\mathrm{P}} u_{\mathrm{P}} = \sum_{\mathrm{nb}} a_{\mathrm{nb}} u_{nb} + S_{\mathbf{u}} \tag{5.5}$$

In these equations a_{P} and a_{nb} are the coefficients accounting for advection and diffusion contributions, $S_{\mathbf{u}}$ is a source term encompassing all contributions not included in the before mentioned coefficients, the subscript P denotes the cell under consideration and subscript *nb* its corresponding neighbor cells.

The set of algebraic equations (Eq. 5.5) are sequentially solved for the Cartesian velocity components by an iterative solver. The newly computed velocity field usually does not satisfy the continuity equation (i.e. Eq. 5.1), thus it needs to be corrected by an adjustment of the pressure gradients that drive it. This is accomplished by means of a pressure-correction field obtained from a discrete Poisson equation, derived from a discretized form of the continuity equation in combination with the momentum equation. This pressure correction equation is also solved by a Jacobi iterative solver. The correction of the velocity field follows the SIMPLE strategy proposed by Patankar (Patankar, 1980). On the SIMPLE iterative procedure, the viscosity is updated at the end of each iteration step using the model given by Eq. 5.3. The developed numerical code uses unstructured meshes and a collocated variable arrangement. For more details on the numerical implementation see Chapter 3.

5.2.2 *CPU and GPU implementations*

With the second generation of GPUs, the Fermi architecture was introduced, and GPUs became more suitable for scientific computations. The GPUs comprises different types of memories, from fast to slow, and with this new architecture memory access is automatic, without the need of programmer's intervention, thus facilitating the implementation of random access memory algorithms.

The flowchart of Fig. 5.1 shows the sequence of tasks for the GPU code implementation, where the white boxes indicate the routines executed on CPU and the gray boxes have the routines executed on GPU. The full CPU implementation is similar to that shown in Fig. 5.1, without the "colouring scheme" that is not required on serial implementations.

To port the numerical code to the GPU, we used the most basic procedure, running on the GPU the most time consuming routines, and minimizing the time required for the data transfer between the CPU and GPU, which is done only at the start and end of the SIMPLE algorithm. Therefore, the mesh generation, data structuring and coloring scheme routines are executed in the CPU, while the heaviest part of the algorithm is ported to the GPU, as can be seen in Fig. 5.1.

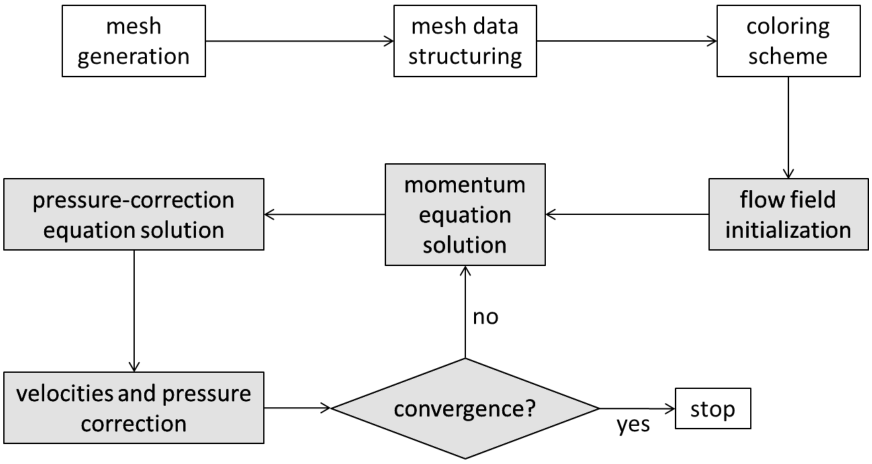

mesh generation
mesh data structuring
coloring scheme
pressure-correction equation solution
momentum equation solution
flow field initialization
no
velocities and pressure correction
convergence?
yes
stop

Figure 5.1: SIMPLE scheme used to compute velocity and pressure fields. The gray boxes represent the routines ported to the GPU.

In order to avoid information loss, that occurs when two or more threads (running in parallel) try to access simultaneously the same memory address, we adopted a colouring scheme (Kampolis et al., 2010) when assembling the systems of equations. Accordingly, we coloured differently those control volume faces that contribute to the same term of the system of equations coefficient matrix diagonal. In this way we guarantee that implicit contributions to that matrix diagonal are not inserted in a concurrent manner.

For both systems of equations (that are solved at each iteration), we chose the point-iterative Jacobi method, which uses a matrix stored in the compressed sparse row format.

The GPU used in the simulations was the NVIDIA G-force GTX480, and the CPU was an Intel Core i7-950 Processor (3.06 GHz) with 8 GB RAM, making use of just one core.

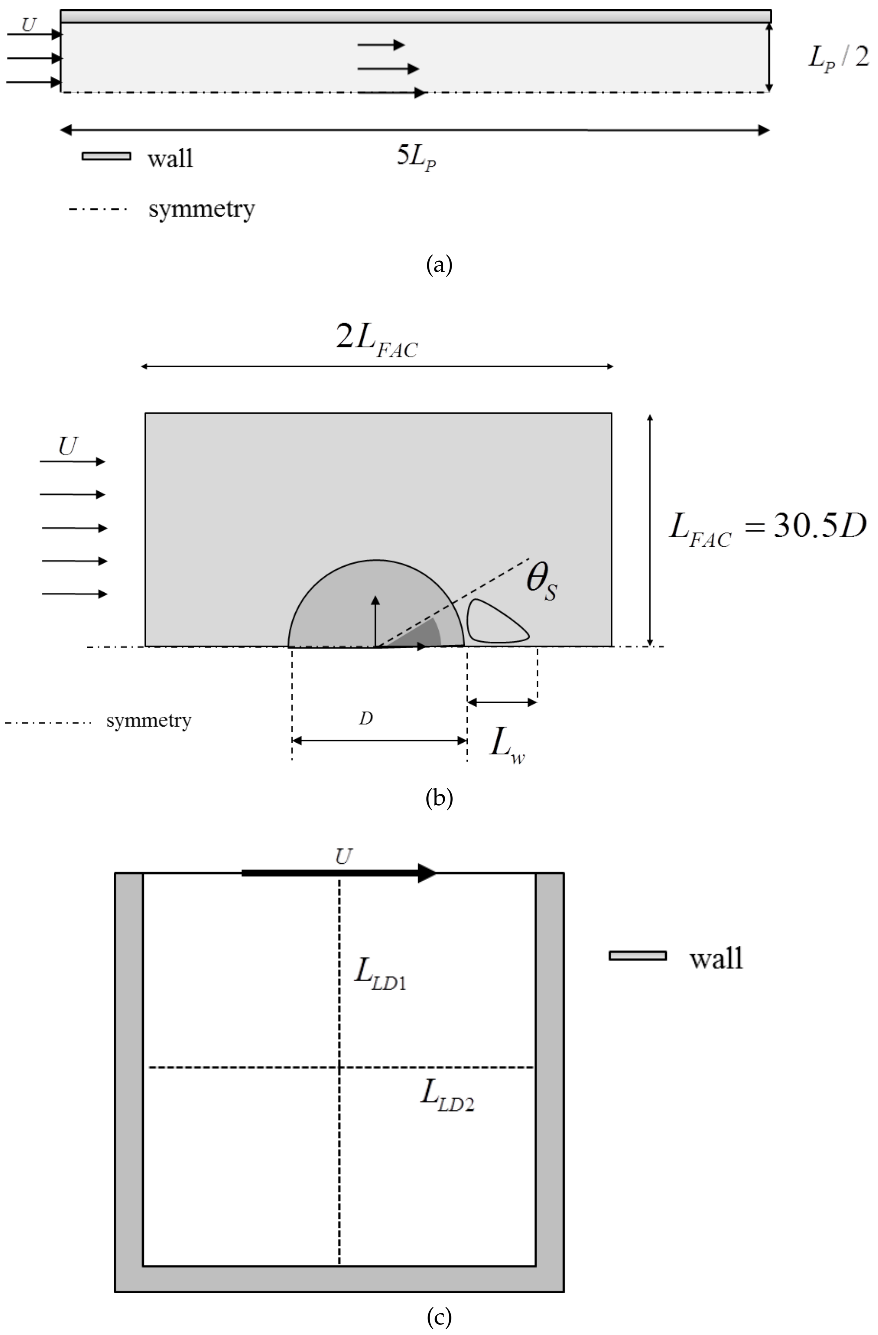
5.3 CASE STUDIES

The results obtained from the numerical solutions, namely Poiseuille flow, lid-driven cavity flow, flow around a cylinder, and flow in a catheter and a wood plastic composite profile dies, will now be presented, together with a speedup comparison between the serial and parallel code implementations. It is important to notice that for all the tested case studies the results and number of iterations obtained with serial and parallel versions of the code were equal.

5.3.1 *Benchmark problems and code verification*

Initially, and in order to verify the code implementation, simple flows like the flow around a cylinder, flow in a simple channel and the lid-driven cavity flow were studied (see Fig. 5.2). Our results were compared with the analytical solutions (Poiseuille flow for $Re = 80$), and with benchmark solutions that exist in the literature for both the flow around a cylinder ($Re = 5$) (Bharti et al., 2006) and the lid-driven cavity flow ($Re = 100$) (Ghia et al., 1982).

In the flow around a cylinder problem, the results were analyzed comparing the length of wake (or recirculation), L_w^*, a dimensionless length, corresponding to the ratio between the length of the recirculation formed on the back side of the cylinder (L_w) and the cylinder diameter (D), and the angle of separation, θ_S, the angle between the symmetry line and the flow separation from cylinder surface (see Fig. 5.2b). The differences, between our results and the ones given in Bharti et al. (2006), obtained for these two parameters, were circa 3.2%, for L_w^*, and 1.7%, for θ_S, with a mesh comprising circa 8 times more cells along the cylinder surface than the one used by Bharti et al. (Bharti et al., 2006), who referred that those results were obtained with an accuracy of circa 1 – 2%.



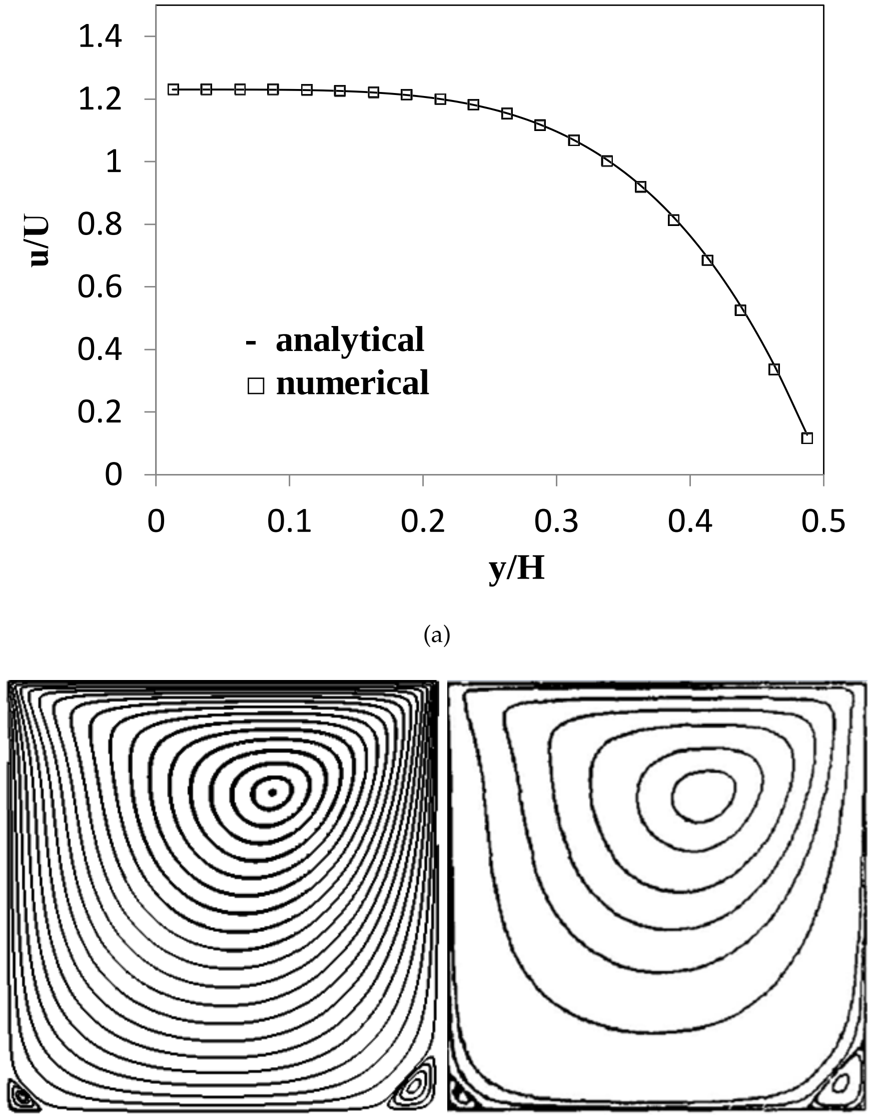
(a)

(b)

(c)

Figure 5.2: Geometries of the case studies used for the code verification and speed up calculations: (a) Poiseuille flow; (b) Flow around a cylinder; (c) Lid-driven cavity flow.

As shown in Fig. 5.3 for the Poiseuille and lid-driven cavity flows, accurate results were obtained, therefore validating our implementations in GPU and CPU. Note that for the lid-driven cavity case study the fluid was assumed to be Newtonian. To validate the code for non-Newtonian fluids, the Poiseuille flow was solved considering a Power-law constitutive equation, assuming the following parameters: $K = 1000\, Pa\, s^n$ and $n = 0.3$.

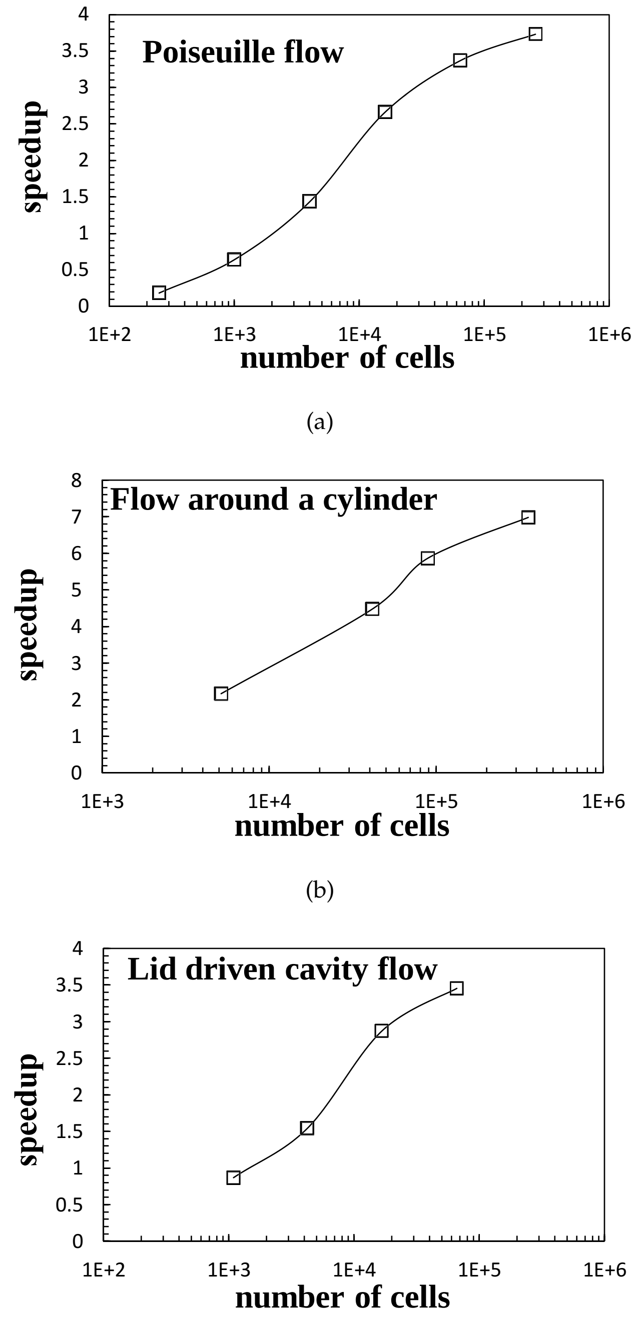


(a)

(b)

Figure 5.3: Results obtained for the case studies employed for the code verification: (a) velocity profile for the Poiseuille flow; (b) streamlines predicted for the Lid-driven cavity flow: left, obtained with the developed numerical code; right, presented by Ghia (Ghia et al., 1982).

In Fig. 5.4 we present the speedups obtained for these three benchmark problems, as a function of the number of cells used in the numerical meshes. As shown, a substantial increase of the code performance is obtained for the three cases. The maximum speedups achieved were 3.7x for the Poiseuille flow, 7x for the flow around a cylinder and 3.5x for the lid-driven cavity flow. It can also be seen that the variation of the speedup with the number of cells leads to a sigmoidal shape, for all cases, evidencing that the performance scales with the mesh size.



(a)

(b)

(c)

Figure 5.4: Speedups obtained for the benchmark case studies: (a) Poiseuille flow; (b) Flow around a cylinder; (c) Lid-driven cavity flow.

Note that for very coarse meshes, the GPU parallel implementation takes more time to perform the simulations than the single CPU. This happens because the time taken by the exchange of information between the CPU and the GPU is comparable to the time consumed to solve the problem. For the most refined meshes the time required for the exchange of information between the GPU and the CPU is residual.

5.3.2 *Design of profile extrusion dies*

The previous results were obtained for quite simple geometries, unlike the complexity of current industrial problems. To design extrusion dies for the production of profiles comprising more complex geometries, extensive experience in the extrusion process is required, as well as the performance of several trials in order to achieve acceptable results. One of the main difficulties on extrusion die design is the achievement of a balanced flow at the flow channel outlet. To overcome these difficulties, numerical codes can be a valuable design aid, allowing to minimize the resources spent on the experimental trial-and-error process. With these tools the designer can improve the extrusion die channel geometry, by using either numerical base trial-and-error procedures (Szarvasy et al., 2000) or automatic algorithms that search for an optimized geometry guided by an objective function, without any user intervention (Nóbrega et al., 2000b).

In order to evaluate the advantage of the employment of the parallelized version of the numerical code on the design of more complex geometries, two additional case studies will be considered, comprising the design of extrusion dies for the production of a medical catheter and a WPC decking profile.

5.3.2.1 *Medical Catheter Extrusion Die*

Medical catheters are devices that can be used to aid the treatment of diseases or the execution of surgical procedures, for instance by facilitating the insertion of drugs or surgical instruments in the patient's body. Catheters can be used on several applications, e.g., cardiovascular, urological, gastrointestinal, neurovascular and ophthalmic. Each application requires a specific catheter that can comprise several holes (lumens) that can possess different diameters. Due to its constant cross section, catheters are produced by extrusion, using medical grade materials.

As shown in Fig. 5.5 (a) the catheter geometry considered on this work has five channels (lumens). Due to symmetry reasons only half geometry was taken into account. It should be noted that the relative location of the channels are not expected to affect the catheter performance, since the lumens functionality is maintained when their diameter is assured (Fig. 5.8 (a)). Thus, the problem to be solved for this profile is to identify the best location of the lumen that ensures the most balanced flow distribution. Accordingly, the geometry was parametrized with the location of the lumen centers (see Fig. 5.5 (b)) (Chapter 3).

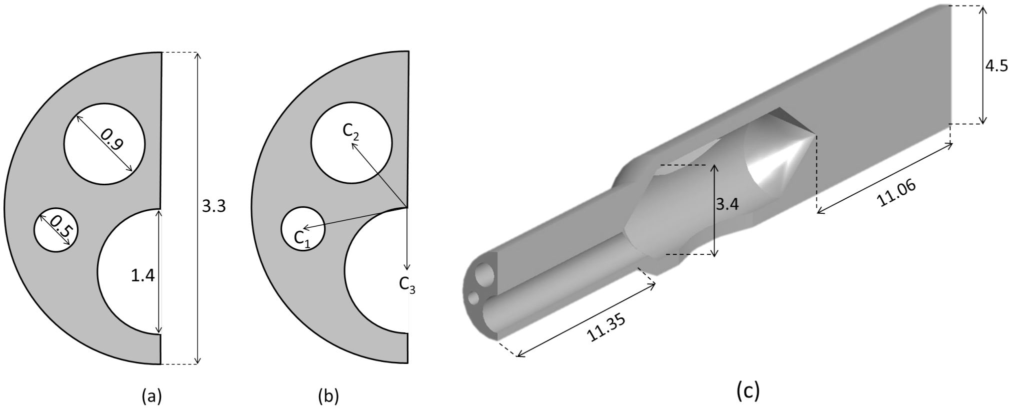


Figure 5.5: Extrusion die geometry for the production of a catheter profile: (a) diameters (mm); (b) location of the channels and (c) flow channel dimensions (mm).

The material employed for the production of the catheter was a polypropylene homopolymer extrusion grade, Novolen PPH 2150, from Targor, which rheological behavior was experimentally characterized in capillary and rotational rheometers, at $230\,^{o}C$ (Nóbrega, Carneiro, Pinho and Oliveira, 2003). The shear viscosity data were fitted to a Bird-Carreau constitutive equation, considering $\eta_\infty = 0\,Pa\,s$, which yield the following parameters: $\eta_0 = 5.58 \times 10^4\,Pa\,s$, $\lambda = 3.21\,s$ and $n = 0.3014$. For the density, a typical value for polypropylene was considered $\left(\rho = 900\,kg\,m^{-3}\right)$.

To evaluate the dependency of speedup between the CPU and the GPU version of the code, regarding to the number of cells, three meshes were used with 12382, 57085 and 389102 cells, being the last illustrated in Fig. 5.6.

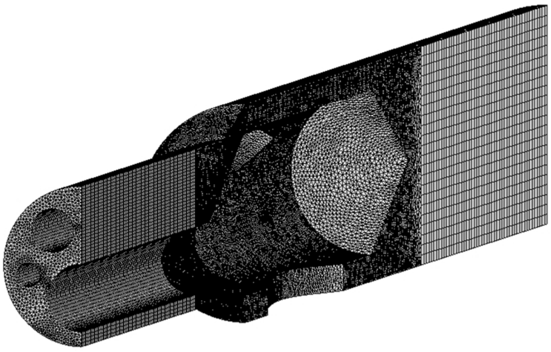


Figure 5.6: Typical mesh used on the medical catheter case study, comprising 389102 cells.

For this specific problem a maximum speedup of 6.4x was obtained (see Fig. 5.7), with the highest value corresponding to the most refined mesh.

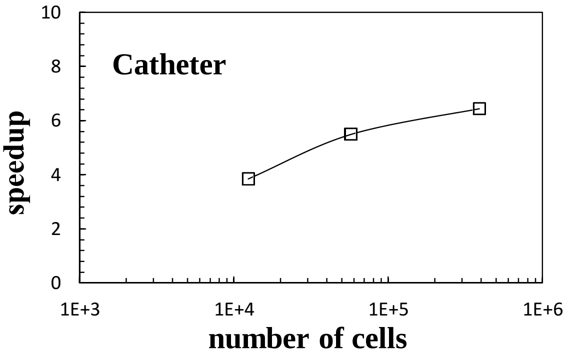


Figure 5.7: Speedup obtained for the medical catheter case study.

The outflow distribution for the initial trial is illustrated in Fig. 5.8 (a); it can be seen that higher values of the velocity occur in regions where the restriction to the flow is lower. In order to balance the outlet flow, several locations for the lumens were tried, being the most balanced geometry obtained on the sixth trial, illustrated in Fig. 5.8 (b). More details on the optimization process are given on Gonçalves et al. 2013.

The improvements obtained during the optimization process can be evaluated by the evolution of the objective function (Chapter 3) calculated for each trial (Fig. 5.9), which value reduces with the improvement of the tool performance. Both results, velocity field and objective function, evidence a significant improvement of the tool effectiveness.

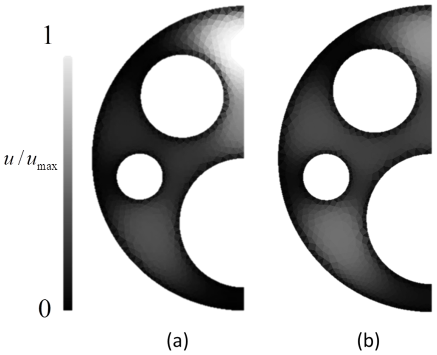


Figure 5.8: Velocity field obtained for the outflow of a catheter profile: (a) initial trial; (b) final trial.

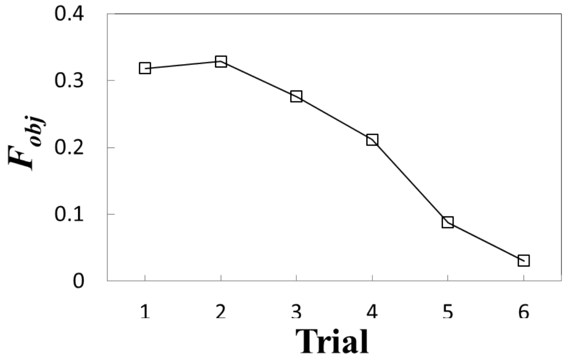


Figure 5.9: Evolution of objective function along the optimization process for the medical catheter case study.

Each run made on the CPU serial code took about 2h50min of computation time, while for the parallelized GPU code the same problem took circa 27 min. Therefore, the full optimization process takes about 17h15min and 2h40min, to run on the CPU and GPU, respectively, i. e., a significant reduction on the total computation time was achieved.

5.3.2.2 *Wood-Plastic Composite Extrusion Die*

The second complex geometry considered in this work is an extrusion die for the production of a wood-plastic composite (WPC) profile. WPCs are mainly made of wood particles dispersed in a thermoplastic matrix. Their main applications lie in the civil construction area, being an alternative to solid wood that requires more maintenance and has less flexibility in terms of geometry.

The dimensions of the cross-section of the initial trial die are shown in Fig. 5.10 (a). Due to symmetry reasons, only half geometry was considered. The achievement of a balanced flow at outlet can be sought, keeping the outside contour shape, and modifying the dimensions of the torpedoes of the extrusion die that shape the hollow sections of the profile (Fig. 5.10 (b)). These changes in the torpedoes can be done easily, since they are removable, and do not affect the profile functionality, which is mainly determined by its outer contour.

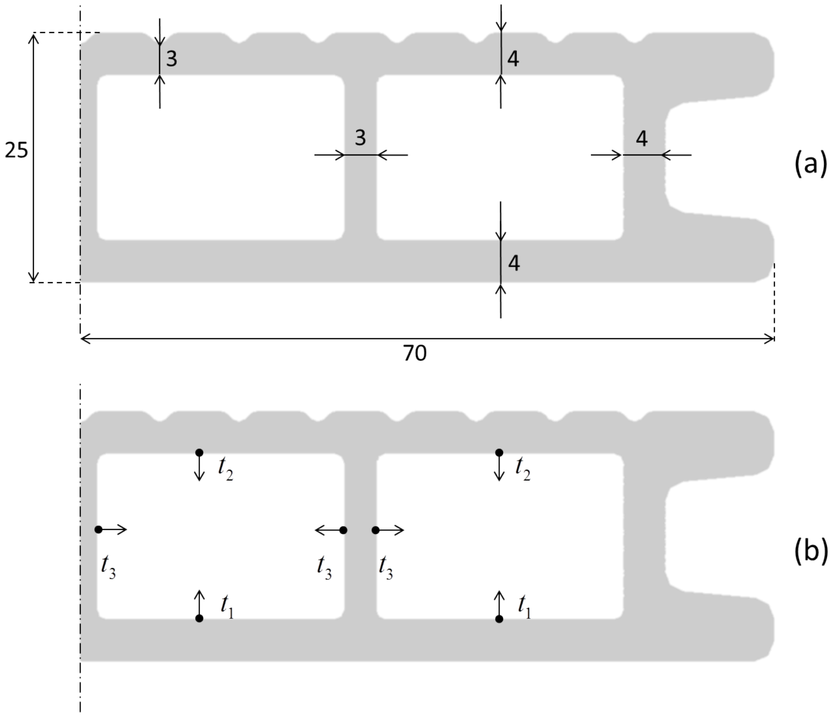


Figure 5.10: Extrusion die geometry for the production of a WPC profile: (a) dimensions (mm); (b) parameters employed for optimization purposes.

To characterize the material, we used a capillary rheometer with dies of 2 mm diameter and L/D of 16 and 4 (L and D stand for the length and diameter of the die, respectively), to perform the Bagley correction. The tests were performed at a temperature of 190ºC. The experimental data (shear viscosity versus shear-rate) were fitted in order to obtain the Bird-Carreau model parameters that resulted in: $n = 0.32$, $\eta_0 = 53993\,Pa.s$, $\eta_\infty = 0\,Pa.s$ and $\lambda = 2.36\,s$. Other characteristics of the WPC are: density $1200\,kg\,m^{-3}$, specific heat $1300\,J\,kg^{-1}K^{-1}$ and thermal conductivity $0.08\,W\,m^{-1}K^{-1}$.

To analyze the evolution of the speedup obtained with the parallelized version of the code relatively to the serial one, as function of the number of cells, three different meshes with 57170, 230282 and 814032 cells, were considered (Fig. 5.11, shows the most refined mesh employed).

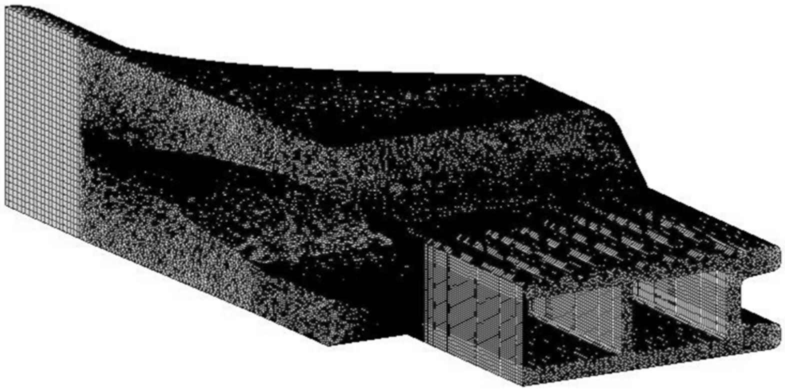


Figure 5.11: Typical mesh used on the WPC case study, comprising 230282 cells.

A maximum speedup of 6.7x was achieved (Fig. 5.12) for the finest mesh employed. However, between the two most refined meshes, there was only a residual increase on the speedup obtained.

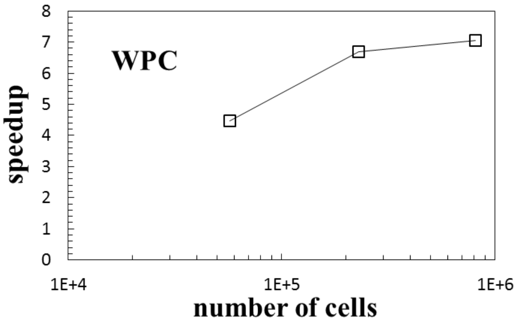


Figure 5.12: Speedup obtained for the WPC profile extrusion die flow simulations.

The flow distribution of the initial trial at the extrusion die outlet cross section is shown in Fig. 5.13 (a), where it can be seen that highest average velocities occur in section where the flow restrictions are lower.

As in the previous case, the flow balance of this die was also optimized, and a similar objective function was used to drive the process. In Fig. 5.14 it can be seen that the objective function value decreases significantly from the first to the last trial.

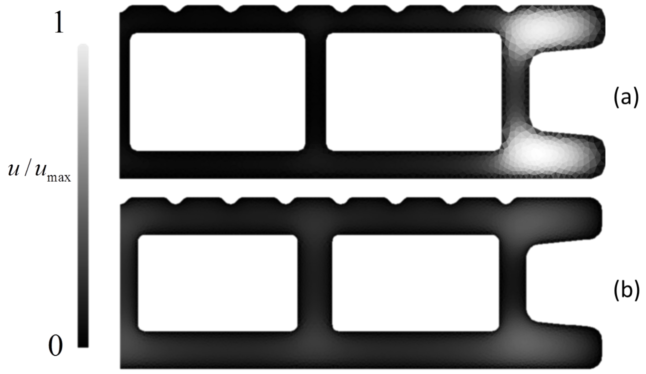


Figure 5.13: Flow distribution at the outlet of the WPC profile extrusion die: (a) initial trial and (b) final trial.

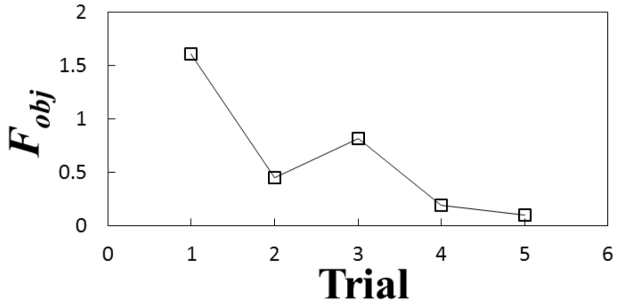


Figure 5.14: Extrusion die for the production of the WPC profile: evolution of objective function along the optimization process.

For this case the computation time needed to each run made on the CPU serial code was about 7h40min, while for the parallelized GPU code the same problem took circa 1h5min. Since the full optimization process needed five runs, it takes about 38h30min and 5h30min, to run on the CPU and GPU, respectively. As in the catheter study case, a significant reduction on the total computation time was achieved with the use of the parallelized GPU code.

5.4 conclusions

This work presents the GPU parallel implementation of a 3D finite volume flow solver for unstructured meshes. The assessment of the code was carried out using three benchmark problems (Poiseuille flow, flow around a cylinder and lid-driven cavity flow), and its ability to deal with complex problems was illustrated with the design of extrusion dies for the production of a medical catheter and of a wood plastic composite profile. In order to evaluate the advantages of the GPU parallelization, speedups between the serial version of the numerical code, that runs on the CPU, and the GPU parallelized numerical code, were computed. For the tested case studies, speedups ranging from 3.5x to 7x were obtained.

In what concerns to the design of complex cross section geometry profile extrusion dies, two case studies were considered. For the design of the medical catheter extrusion die, six numerical runs were required to attain an acceptable flow balance. The computation time required decreased from 17 h and 15 min to 2 h and 40 min on the serial (CPU) and parallelized (GPU) implementations of the numerical code, respectively. For the wood plastic composite decking profile case, the 5 runs required for the design process took 38 h and 30 min and 5 h and 30 min, on CPU and GPU implementations of the numerical code, respectively.

From the results obtained it can be concluded that the GPU parallelization of the numerical code allowed a significant reduction of the time spent in calculation, which will have a noticeable positive impact on the design process. It is important to notice that this results were obtained without any complex memory management on the GPU and, therefore, there is room for additional future improvements.

6

SIMPLIFIED GUIDELINES FOR THE DESIGN OF COMPLEX PROFILE EXTRUSION DIES

Abstract[1]

In this work the influence of the parallel zone length and thickness on the extrusion die flow distribution is investigated numerically, with a view to establish useful guidelines to help those who are involved on the design of extrusion dies and do not have access to adequate numerical modelling codes. The simulations were performed using a computational rheology code based on the Finite Volume Method that is able to predict the flow of temperature-dependent inelastic non-Newtonian fluids on complex geometries. For the viscosity dependence on the shear-rate and temperature, the Bird-Carreau constitutive equation and the Arrhenius law were employed, respectively. Two elementar cross section shapes were studied, namely a L-shaped and a T-shaped, due to their relevance on the definition of the most common extruded profiles geometry. For the L-shaped die an exponential expression was found relating the thickness ratio and the length of the pre-parallel zone needed to achieve a balanced flow distribution, whereas for the T-shaped die a quadratic expression fits better this relation.

6.1 INTRODUCTION

The need to control and understand "how a fluid flows" goes back to the early ages. The most natural/simple example is the deviation of rivers from their natural course, so that plantations could receive the proper amount of water, or, for preventing

[1] N. D. Gonçalves, L. L. Ferrás, O. S. Carneiro and J. M. Nóbrega (2013). Simplified guidelines for the design of complex profile extrusion dies, in preparation for submission in Journal of Material Processing Technology

houses and civilizations from being flooded. With the Industrial Revolution several new industrial branches have emerged, and the demand for fluid flow control evolved from a simple basic need to an everyday indispensable process, involving complex flows and different types of fluids. One of those branches is the polymer processing industry, where the need to understand and model the fluid flow is essential in several processes, such as extrusion (of films, fibers, filaments, pipes, wire coating and profiles), injection molding, thermoforming and blow molding, to name just the most common ones.

In this work we will focus on the extrusion process, in which the extruder converts a solid pelletized feedstock into a homogeneous melt and pumps it at a constant rate. The polymer, in its molten state, is then shaped, in an extrusion die, being subsequently cooled down to its solidification temperature in a calibration system, in order to preserve its shape along the final process stages, storage and service. The polymer melt shaping stage involves shear and elongational deformations in a flow channel that comprises uneven flow restrictions, therefore contributing to the difficult task of obtaining a homogeneous output velocity field, i. e., a balanced flow distribution (Nóbrega, Carneiro, Pinho and Oliveira, 2003; Michaeli and Kaul, 2004).

The main problem to solve when designing a profile extrusion die is how to promote the required flow distribution, i. e., to guarantee a similar average velocity of the melt for all the sections of the extrudate, at the die exit. For that, we need to adequately adjust the extrusion die flow channel geometry, taking into account that a small geometrical change will influence the overall velocity field distribution. An experimental trial and error procedure is a possible, and commonly employed, method to obtain an adequate the solution, but it is time and material consuming and the performance of the final extrusion die is highly dependent on the designer's experience.

The literature is rich on experimental and numerical studies involving the optimization of extrusion dies flow channel geometry. Regarding experimental studies we have the early work of Carley (1954) where he analyzed the laminar flow of thermoplastic polymers in crosshead-slit dies (used for the extrusion of sheet, flat film and blown film). He derived an equation relating pressure to position along the slit for materials that obey the Power Law model. The devised methodology allowed to quantify the uniformity index (UI), and he concluded that the UI for center-fed dies was always greater than for end-fed dies of the same length. There is also the work of Pearson (1964), where he investigated the flat-film die design for non-Newtonian fluids. The interested reader is referred to the book of Carneiro and Nóbrega (2012).

On the branch of numerical studies, McKelvey and Ito (1971) presented a method for analyzing the flow uniformity in flat sheet dies, assuming a Power Law model. They also employed the concept of UI, presented methods to establish the die dimensions, to obtain an even flow distribution at particular flow rates and pressure drops and concluded that the Power Law index "n" is the system parameter that has the highest influence on the flow distribution. Proctor (1972) presented an analysis of inelastic Power Law fluids under isothermal conditions and showed that the axial flow distribution in spiral-mandrel and side fed film and flat sheet extrusion dies were critically dependent upon a number of flow channel design parameters, specially on the Power Law index. Later, Chung and Lohkamp (1976) demonstrated using an analytical approach that the melt flow through coat-hanger extrusion dies can be approximated by simplifying the flow path and using the Power Law model. Tadmor et al. (1974) presented the flow analysis network, a method for solving flow problems in polymer processing. Basically, they used a Finite Element Method (FEM) based code for solving two dimensional flow problems in complex geometrical configurations. Wortberg et al. (1982) investigated the three main stages of the extrusion process: extruder, die forming and cooling. They calculated the three dimensional flow patterns in extrusion dies by means of the FEM, for Y and T shaped profiles.

Legat and Marchal (1993) presented a mathematical formulation for the solution of the inverse problem, i. e., finding the die cross section geometry which produces an extrudate with the prescribed shape, that minimizes the "trial-and-error" iterative phase on the definition of the die geometry. In this field several other works have been proposed in the literature, mostly based on FEM and using sequential optimization methods, genetic based optimization algorithms, adaptable die design and sensitivity analysis (Hurez et al., 1996; Chen et al., 1997; Na and Lee, 1997; Yu and Liu, 1998; Ulysse, 1999, 2002; Lehnhäuser and Schäfer, 2005; Michaeli et al., 2001; Michaeli and Kaul, 2004; Sienz et al., 2006; Lebaal et al., 2009; Soury et al., 2009; Gordon et al., 2007; Sienz et al., 2010; Mu et al., 2010; Elgeti et al., 2012). For an overview of some of the breakthrough inventions and innovations in die design we advise the work of J. Dooley (2009).

Most of these works make use of the FEM. Works using the Finite Volume Method (FVM) are more scarce, but some studies can be found in the literature such as the ones of Carneiro and Nóbrega (Carneiro et al., 2001; Carneiro, Nóbrega, Oliveira and Pinho, 2003; Nóbrega, Carneiro, Pinho and Oliveira, 2003, 2004), that present an automatic process for the optimization of an extrusion die able to deal with simple profiles. Their method was based on a "numerical trial and error" procedure, mimicking the experimental one used by die designers (Carneiro and Nóbrega, 2012).

With the recent advances in the computational field, the use of numerical tools to predict and correct the fluid flow distribution is well established as a good alternative to the exclusive use of the experimental approach, reducing the costs involved and the number of experimental trials required. However, despite the usefulness of the numerical techniques presented, a general tool, able to deal with any die geometry and being independent of the designer experience, is still far beyond the available knowledge and capabilities. Furthermore, many companies have no access to any numerical tools. Therefore, it would be useful to investigate, identify and understand the possible relations that may exist between the fluid flow distribution and the die dimensions that result in an homogeneous flow field distribution, aiming the establishment of useful guidelines for those that do not have access to adequate numerical modeling tools.

The objective of this work is then to contribute to the establishment of the above mentioned relations. For that we use a numerical modeling code described in Gonçalves et al. (2013) that is able to predict the flow of temperature dependent inelastic non-Newtonian fluids, and study the behaviour of two elementar geometries, L-shaped and T-shaped dies, since a large range of more complex geometries can be decomposed into these two simple geometries.

The remaining of this chapter is organized as follows. In Section 6.2 the design methodology and the geometry used in the simulations are presented. The numerical modelling code is briefly described in Section 6.3. In Section 6.4 the optimization results are presented and discussed and the chapter ends with the conclusions in Section 6.5.

6.2 problem statement

Different wall thicknesses, usually present in complex profiles, originate different flow restrictions that have a major impact on the obtained flow distribution at the outlet of the extrusion die flow channel. If the tool is not properly designed, the thicker sections will have higher average flow velocities, and vice-versa, which hinders the production of the desired profile (see Fig. 6.1).

For design purposes only the ESs parameters can be adjusted; the flow in the Intersections (Is) can not be directly controlled, being, thus, a consequence of the flow distribution in the neighbor ESs.

A typical profile extrusion die is defined by four main zones (see Fig. 6.2): i) an adapter, connecting the extruder discharge zone to the die; ii) the transition zone performing the transition from the adapter to the pre-parallel zone; iii) a pre-parallel zone, with a similar geometry to that of the parallel zone, connecting the transition zone and the parallel zone, used to control the flow distribution at the die exit; iv) a final parallel zone of constant cross section where the fluid acquires its final shape (Nóbrega, Carneiro, Pinho and Oliveira, 2003; Carneiro and Nóbrega, 2012). For a given profile cross section geometry the evaluation of the flow distribution should be done considering the average outlet velocity in each ES, which should be the same for all ESs. To control the local flow restriction, adjustments can be made on the controllable geometrical parameters associated to each ES, illustrated in Fig. 6.3 , that allow to define the flow channel geometry: parallel zone length, or length of constant

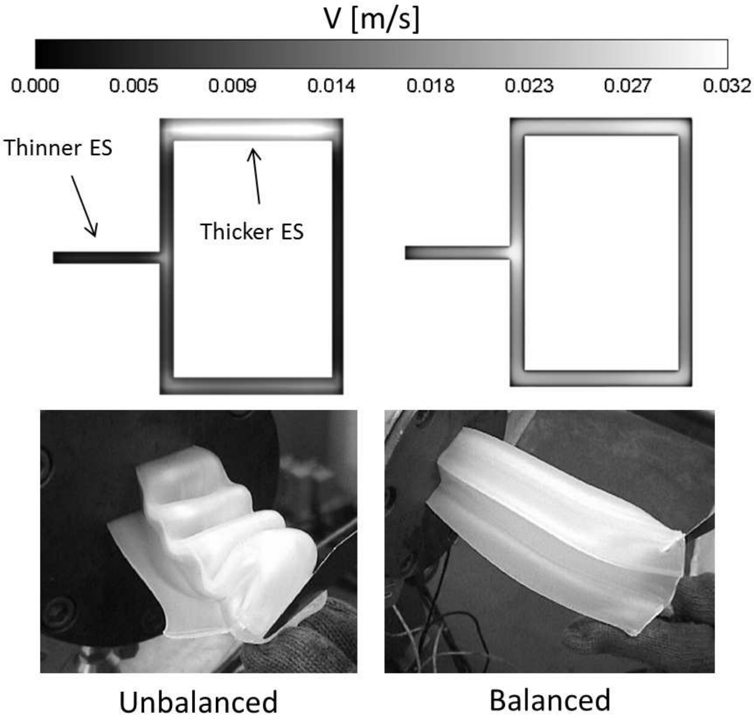


Figure 6.1: Velocity contours obtained at the flow channel outlet and produced extrudate both for an unbalanced and balanced extrusion dies (adapted from Nóbrega (2004)).

thickness, (L), angle of convergence and compression ratio (Nóbrega, Carneiro, Pinho and Oliveira, 2003).

As mentioned before, in this work the behaviour of extrusion die for the production of two elementar cross section geometries, L-shaped and T-shaped, will be studied. Both cross sections comprise two ES, and the work done aims at determining the parallel zone length to thickness ratio (L/t) required to balance the flow at the flow channel outlet, and how the thickness ratio (i. e., ratio of the two ES thicknesses) affects that value.

As a general rule, to obtain a even flow distribution, the ESs that have an higher restrictive cross section (smaller thickness) should have shorter parallel zone lengths (see Fig. 6.2). However, it should be mentioned that in cases where the difference between ES thicknesses is excessively high, the even flow distribution can not be achieved with this methodology, since above a certain length additional increases in

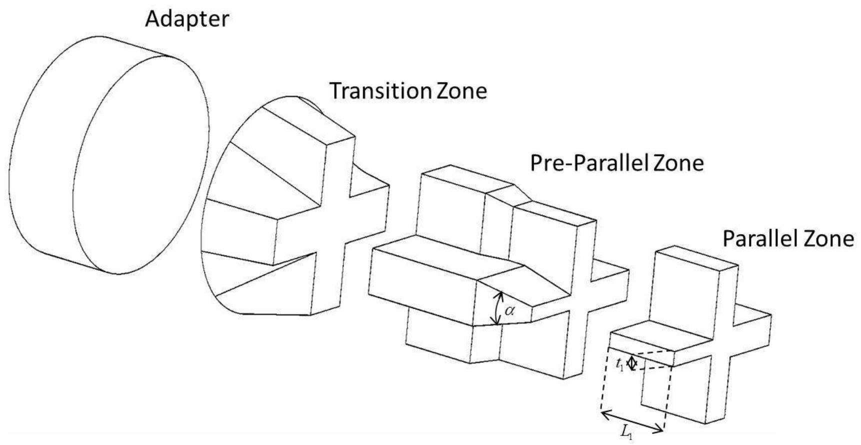


Figure 6.2: Typical extrusion die flow channel, for the production of a profile, its division into different zones and identification of the conttrolable geometrical parameters.

this parameter will have no effect on the flow distribution, because the flow achieves a fully developed condition.

6.2.1 *Numerical modelling code*

The non-isothermal incompressible fluid flow considered in this work is governed by the continuity,

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{6.1}$$

momentum,

$$\rho\left(\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\eta\left(\dot{\gamma}, T\right)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{6.2}$$

and energy,

$$\rho c_P\left(\frac{\partial T}{\partial t} + \frac{\partial u_j T}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left[k\frac{\partial T}{\partial x_j}\right] + \eta\left(\dot{\gamma}\right) D^2_{ij} \tag{6.3}$$

equations, where t is the time, x_i the i^{th} position component, u_i is the i^{th} velocity component, p is the pressure, T is the temperature, ρ is the fluid density, c_p is the specific heat, k is the thermal conductivity, and $\eta\left(\dot{\gamma}, T\right)$ is a variable shear viscosity that depends both on the shear rate $\dot{\gamma}$ and temperature, $\left(\dot{\gamma} \equiv \sqrt{2tr\mathbf{D}^2}, \mathbf{D} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right)$. The last terms on the right-hand-side of Eq. 6.3 accounts for viscous dissipation.

The shear viscosity, $\eta\left(\dot{\gamma}, T\right)$, was modeled using a Bird-Carreau model with an Arrhenius temperature dependence given by

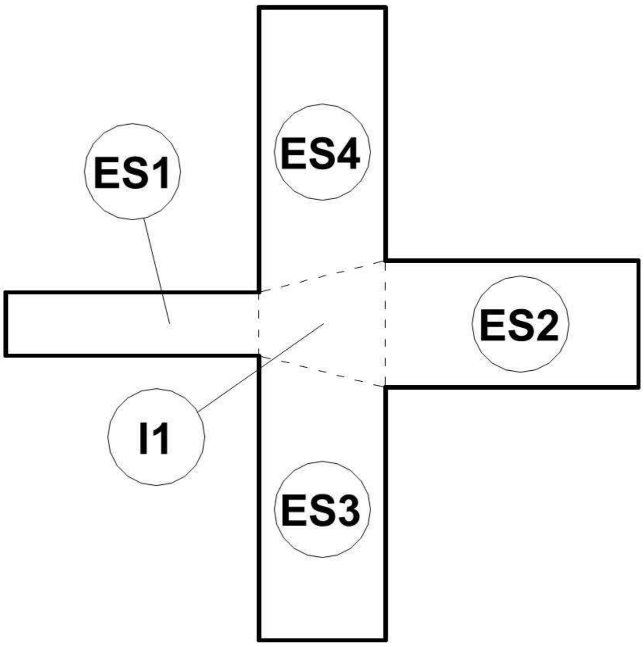


Figure 6.3: Example of the outlet cross section division in Elemental Sections (ESs) and resultant intersection (I).

$$\eta\left(\dot{\gamma},T\right)=a_T\eta_\infty+\frac{a_T\left(\eta_0-\eta_\infty\right)}{\left(1+\left(a_T\lambda\dot{\gamma}\right)^2\right)^{\frac{1-n}{2}}} \tag{6.4}$$

where the temperature shift, a_T, is given by,

$$a_T=exp\left(\frac{E}{R}\left(\frac{1}{T}-\frac{1}{T_{ref}}\right)\right). \tag{6.5}$$

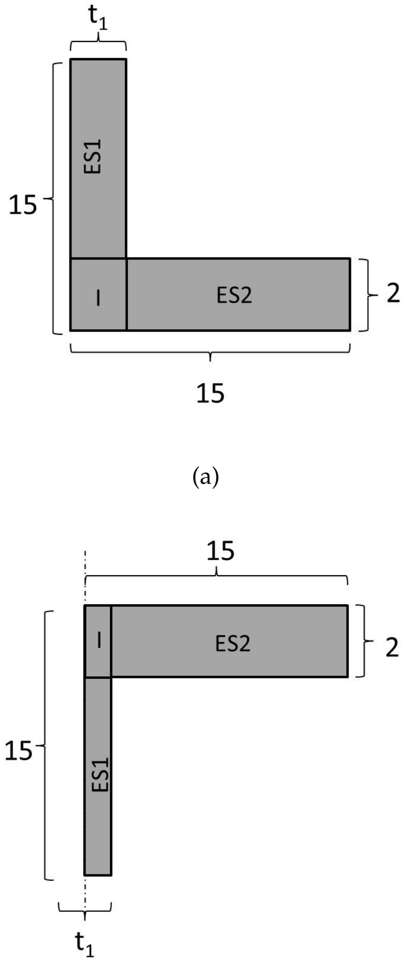
In Eq. 6.4 and Eq. 6.5, η_0 is the zero-shear-rate viscosity, η_∞ is the infinite-shear-rate viscosity, λ is a characteristic time constant (relaxation time), n is the power-law index, E is the activation energy, R the universal gas constant and T_{ref} is the reference temperature for which $a_T=1$.

At the die inlet and channel walls the velocity and temperature fields were prescribed, being the no-slip condition used for the velocity at the walls. At the die outlet null gradients in the normal direction were considered.

To solve this system of equations (Eqs. 6.1, 6.2 and 6.4) we used a numerical code based on the FVM able to deal with unstructured meshes, with a second order space discretization and collocated variables arrangement. Momentum and continuity equations were coupled using the SIMPLE method (Patankar and Spalding, 1972) with the viscosity, $\eta\left(\dot{\gamma}, T\right)$, being updated at the end of each iteration (for more details see Gonçalves et al. (2013)).

6.2.2 *Case Studies*

As mentioned above, a large number of complex geometry profiles can be formed by a combination of two simple geometries: L-shaped and T-shaped. By this reason, in this work, these two simple geometries are studied, in order to establish relations between the geometrical parameters needed to balance the flow distribution and their ES thicknesses ratios.



(a)

(b)

Figure 6.4: Cross section of the flow channel parallel zone and division into ES and I: (a) die-L and (b) die-T (dimensions in mm). Due to symmetry reasons only half of the cross section was considered for die-T.

To monitor the flow distribution, the outlet cross section of both geometries was divided into two rectangular ESs (ES1 and ES2) and a rectangular intersection zone (I) (Fig. 6.4 (a) and (b)). Notice that because of symmetry only half of the die-T flow channel was considered for modelling purposes.

Figs. 6.4 (a) and (b) shows the cross section geometry of the L-shaped and T-shaped dies parallel zone, named respectively as Die-L and Die-T. In order to perform the study for several values of the thicknesses (t_1/t_2), i. e. ratio between ES1 (t_1) and ES2 (t_2) thickness, t_2 was kept constant, equal to $2\,mm$, being changed just the value of t_1. The numerical tests performed covered a range of t_1/t_2 from 0.375 to 0.875.

The Die-L and Die-T flow channels conceived presented in Fig. 6.5, where the main geometrical parameters are illustrated. For both dies and all the studies, the length L_2 was also kept constant $(L_2 = 30\,mm)$, a values high enough to obtain a fully developed flow (Carneiro and Nóbrega, 2012). This means that in this study, for each specific value of t_1/t_2 we search for the L_1/t_1 value required to obtain a balanced flow distribution.

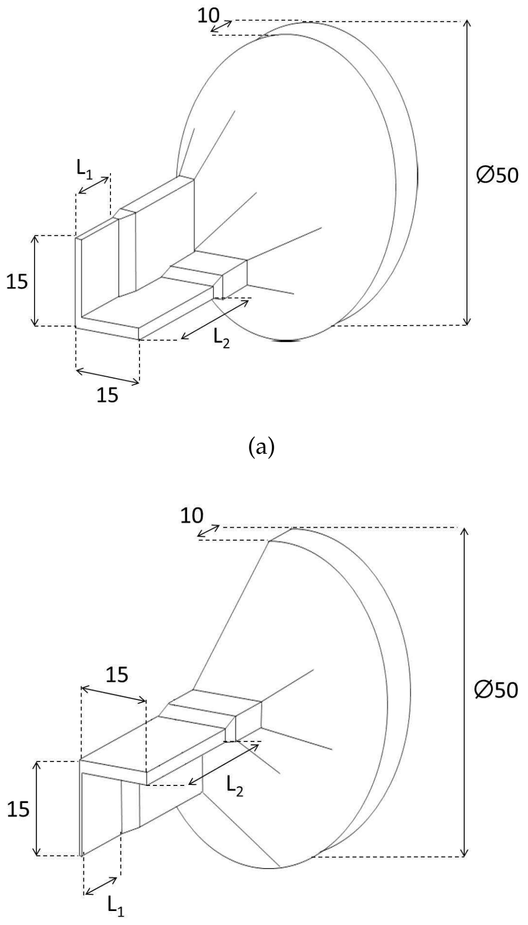
The studies to be performed should also allow to identify, for both dies, the limit values of the thicknesses ratio (t_1/t_2), bellow which a balanced flow cannot be achieved just by changing the L_1, the ES1 length.

As proposed by Nóbrega et al. (Carneiro et al., 2001; Carneiro, Nóbrega, Oliveira and Pinho, 2003), the quality of the flow distribution was quantified by the following objective function:

$$F_{obj} = \sum_{i}^{n_{ES}} \left|\left|1 - \frac{\overline{V_i}}{\overline{V}}\right|\right| \frac{A_i}{A} \qquad (6.6)$$

where $\overline{V_i}$ is the average velocity of section i, $\overline{V}$ the global outlet average flow velocity, A_i is the area of section i, A the total outlet area and n_{ES} the total number of sections (ES and I). Defined this way the F_{obj} is always positive and tends to zero as the quality of the flow distribution is achieved.

The polymer used in this work a Novolen PPH 2150 from Targor (a polypropylene homopolymer extrusion grade). The parameters required to account for the shear viscosity dependence on shear rate and temperature (Eq. 6.4 and Eq. 6.5) were calculated from a least square fit to the experimental data obtained using capillary



(a)

(b)

Figure 6.5: 3D overview of the flow channel: (a) die-L and (b) die-T (dimensions in mm). Due to symmetry reasons only half of the cross section was considered for die-T.

and rotational rheometers, at $210\,^oC$, $230\,^oC$ and $250\,^oC$ (Nóbrega, Carneiro, Pinho and Oliveira, 2003). The calculated model parameters for the Bird Carreau constitutive equation (Eq. 6.4) were the following: $n = 0.3014$, $\eta_0 = 55800\,Pa\,s$, $\eta_\infty = 0\,Pa\,s$, and $\lambda = 3.21\,(s)$. For the Arrhenius law (Eq. 6.5) the experimental parameters values obtained were the following: $T_{ref} = 230\,^oC$ and $E/R = 2900\,K$ (Nóbrega, Carneiro, Pinho and Oliveira, 2003).

In terms of boundary conditions, the temperature was set to $230\,^oC$ at the inlet and walls of the die flow channel, and the inlet velocity was set in order to result in an outlet average velocity of $3\,m/min$, a typical value used in profile extrusion (Carneiro and Nóbrega, 2012).

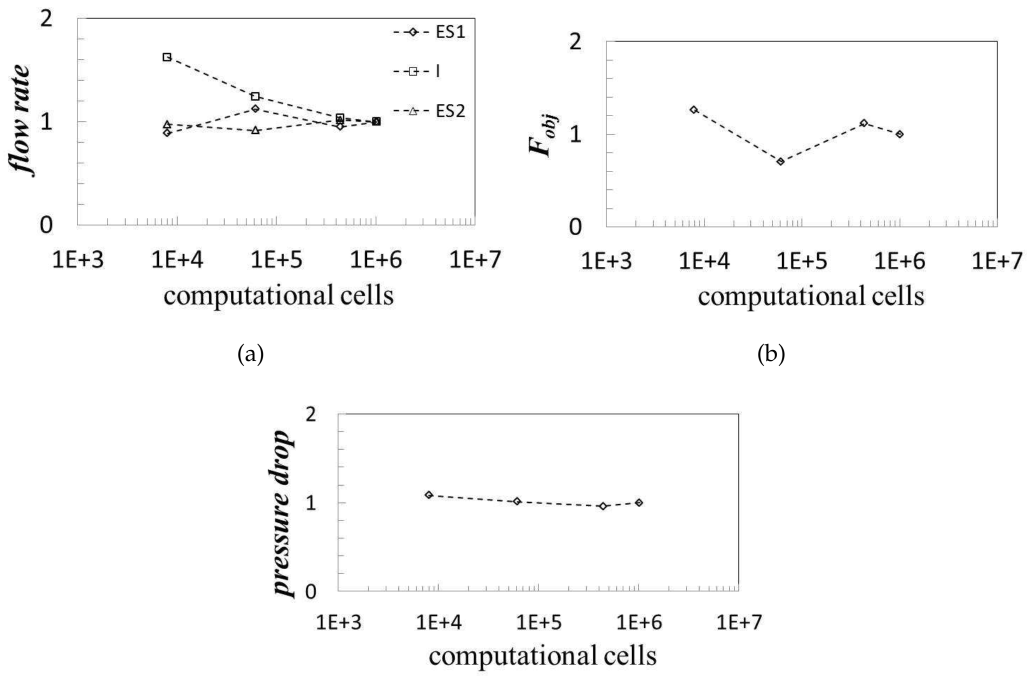
The unstructured meshes used in the simulations are composed by triangular prisms on the flow channel parallel zone, and tetrahedrons elsewhere.

To define the appropriate mesh refinement level, the die-L geometry, with $t_1 = 1.5\,mm$ and $L_1/t_1 = 15$, was modeled with four meshes with increasing number of computational cells (8000, 61416, 439815 and 1015499). In Fig. 6.6 the graphs of the relative values of some relevant results, considering as reference value the one obtained with the finer mesh, are presented for all the meshes. It can be seen in Fig. 6.6 (a) that the individual flows in ES1, I and ES2, converge to the values obtained with the finer mesh, being the differences between the two finer meshes quite small. The same analysis was done for the objective function (Fig. 6.6 (b)) and pressure drop (Fig. 6.6 (c)) where similar trends were observed. Based on these results, the remaining simulations were based on meshes whose refinement level is similar to those of the mesh with 439815 elements. based on this results, for the die-L, with $t_1 = 1\,mm$ and $L_1/t_1 = 15$, a mesh with 110831 tetrahedrons and 229910 triangular prisms, which results in a total of 340741 cells, was used (see Fig. 6.7). For the die-T (with similar dimensions of those of the die-L) the employed mesh has 107773 tetrahedrons and 170826 triangular prisms (278599 cells). Considering these meshes, and based on the fact that there are five unknowns for each computational cell (u_1, u_2, u_3, p and T) the total number of degrees of freedom is 1703705 and 1392995, respectively, die-L and die-T.

6.3 RESULTS AND DISCUSSION

The study comprised numerical runs for both dies. Figs. 6.8 and 6.9, show the effect on the dimensionless average velocity, given by the $\left(\frac{\overline{V_i}}{\overline{V}}\right)$ ratio (where $\overline{V}$ stands for the global outlet average velocity and $\overline{V_i}$ for the average velocity in ES_i) as a function of the L_1/t_1 ratio, for die-L and die-T, respectively.

For die-L, it can be seen in Fig. 6.8 (b) that for a value of $L_1/t_1 = 1$, the ES1 average velocity is higher than the ES2 one. With the increase of L_1/t_1, the opposite situation is reached. From the same figure, one can see that the value of L_1/t_1 that promotes a



(a)

(b)

(c)

Figure 6.6: Results obtained on the mesh sensitivity studies. Ratio between the values obtained with each mesh and those obtained with the finer mesh for the: (a) flow rate; (b) objective function; (c) pressure drop.

similar average velocity on ES1 and ES2 should be achieved L_1/t_1 for values between 1 and 3. Therefore, to the intersection/balance point between ES1 and ES2 velocities can be obtained by interpolation.

For the same die, we see in Fig. 6.8 (a) that no intersection could be found between the ES1 and ES2 average velocities. This occurred because of the ES1 small thickness, $t_1 = 0.75\,mm$, which promoted a very high flow restriction. Thus even for lower values of L_1 (small values of L_1/t_1), the restriction could not be compensated (notice the small average velocity values obtained for all L_1/t_1 values in ES1). With the increase of the t_1/t_2 ratio, the value L_1/t_1 required to obtain the same average velocity on ES1 and ES2 also increase (see Figs. 6.8 (b) to (e)). These results show that a balanced flow distribution could only be achieved when t_1 is equal or higher than $1\,mm$, which corresponds to a thickness ratio, t_1/t_2, equal or higher than 0.5.

For die-T, the results were qualitatively similar to the ones obtained for the die-L. Again, for very small ES1 thickness values a balanced flow condition could not be achieved (see Fig. 6.9 (a) for $t_1 = 0.75\,mm$). Such balance could only be found when

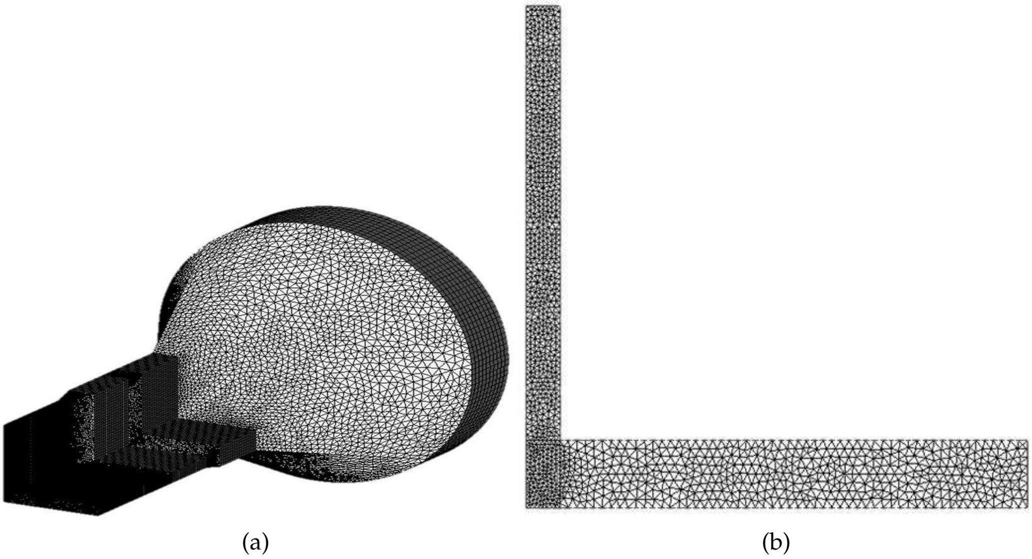


Figure 6.7: Typical mesh used for the die-L geometry: (a) flow channel; (b) cross section of the flow channel parallel zone.

the thickness of the thinner ES is equal or higher than $1\,mm$, which corresponds again to a thickness ratio, t_1/t_2, equal or higher that 0.5 (see Figs. 6.9 (b) to (e)). Simulations with smaller values of t_1 were also performed, but due to the higher restriction imposed, the balanced flow condition was never achieved.

In order to devise simplified guidelines to aid the design of more complex geometry profile extrusion dies, functions relating the flow balance point as function of t_1/t_2 ratio were fitted for both dies. As illustrated in Fig. 6.10, an exponential increase of L_1/t_1 with t_1/t_2 was identified for die-L, for which the regression equation is $L_1/t_1 = 0.0339e^{7.2182(t_1/t_2)}$, with $R^2 = 0.9991$. Similarly, for the die-T, the best fit was achieved with a quadratic function (see Fig. 6.10), for which the regression equation is $L_1/t_1 = -39.88\,(t_1/t_2)^2 + 98.305\,(t_1/t_2) - 37.444$, with $R^2 = 0.9976$.

For illustration purposes, the velocity contours at the extrusion die flow channel outlet cross section, for balanced and unbalanced geometries, is shown in Fig. 6.11 (a) and (b) for the die-L and die-T, respectively. As expected, the balanced flow distribution resulted in a more homogeneous velocity field at the extrusion die outlet (except in I section, that is not directly controlled), for both geometries. The outlet temperature distribution for the same geometries is also plotted in Fig. 6.11, where it is shown that a smaller temperature gradient is obtained for the balanced geometries, when compared with that obtained for the unbalanced ones.

Aiming to evaluate the effect of the material rheological data on the devised guidelines, the influence of the Power Law index, *n*, on the output flow distribution was also analyzed. As shown in Fig. 6.12, for both dies, the value of *n* slightly affects the conditions required to obtain a balanced flow distribution, which confirms the generality and usefulness of the obtained relations. However, aiming to enlarge the applicability of the proposed guidelines, these results show that this study should be extended, in future works, to consider other rheological and/or processing parameters.

6.4 CONCLUSIONS

In this work simple guidelines for the design of profiles extrusion dies were devised. For this purpose, extrusion dies for the production of L and T shaped profiles, that when combined can generate a large range of commonly employed extruded profiles, were numerically studied. The aim was to identify relationships between the flow channel geometry that promoted a balanced flow distribution, and the differential flow restrictions generated by the two elemental sections comprising the studied profiles cross sections. The results obtained allowed to identify polynomial relations between the elemental sections thicknesses ratio and the length over thickness ratio of the most restrictive section. Quadratic and exponential relations were obtained for the T and L shaped dies, respectively. These polynomial relations provide a simple way to establish the flow channel initial trial geometry for extrusion dies used to produce complex geometry profiles , formed by L and T sections. These tools are, therefore, intended for those who do not have access to numerical modelling codes, and thus are expected to allow significant savings on the trial-and-error procedure usually employed on the design of complex extrusion dies.

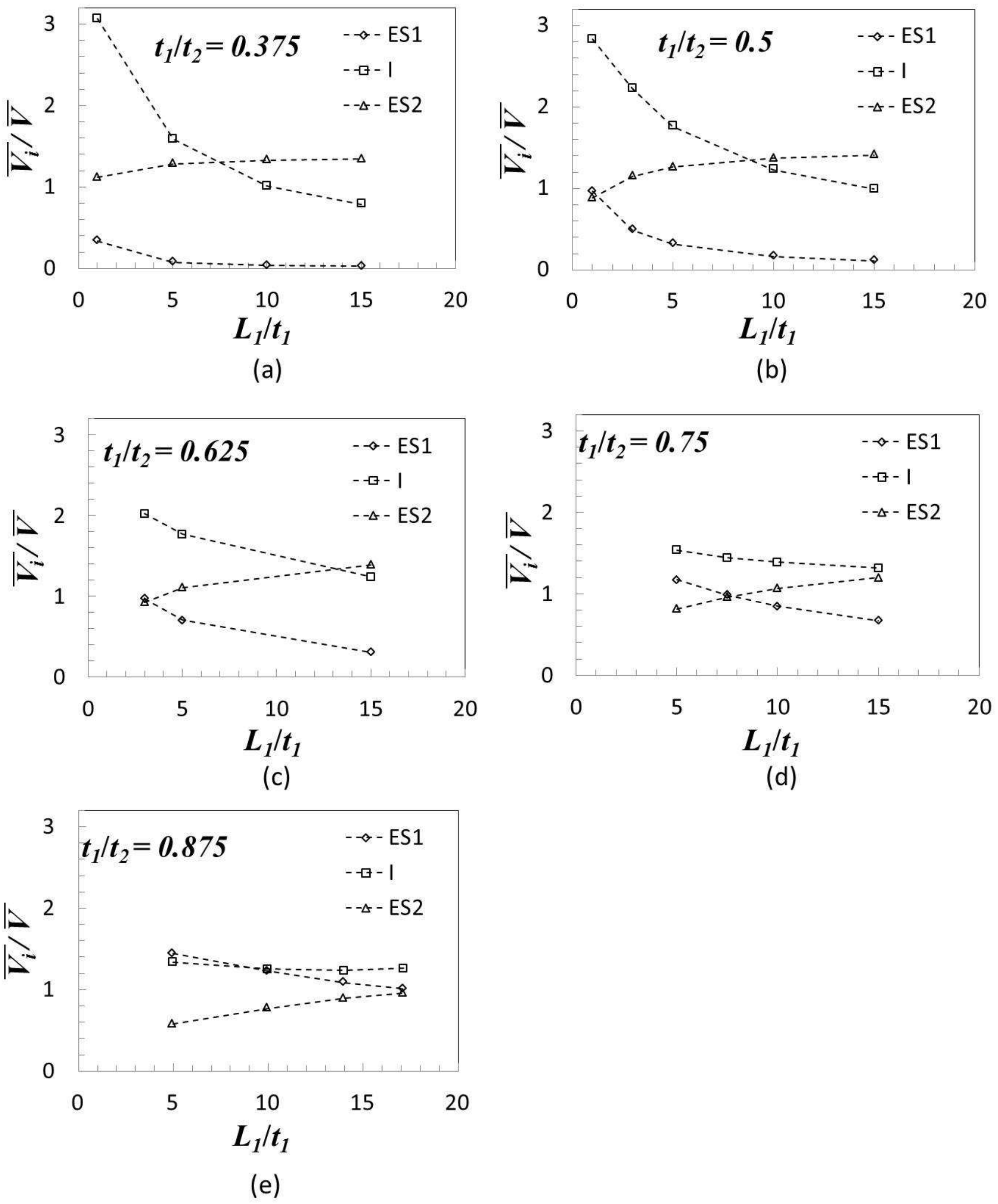


Figure 6.8: Effect of the length/thickness ratio of ES1 (L_1/t_1) on the normalized average velocity $\left(\frac{\overline{V_i}}{\overline{V}}\right)$ for die-L geometry with: (a) $t_1/t_2 = 0.375$; (b) $t_1/t_2 = 0.5$; (c) $t_1/t_2 = 0.625$; (d) $t_1/t_2 = 0.75$; (e) $t_1/t_2 = 0.875$.

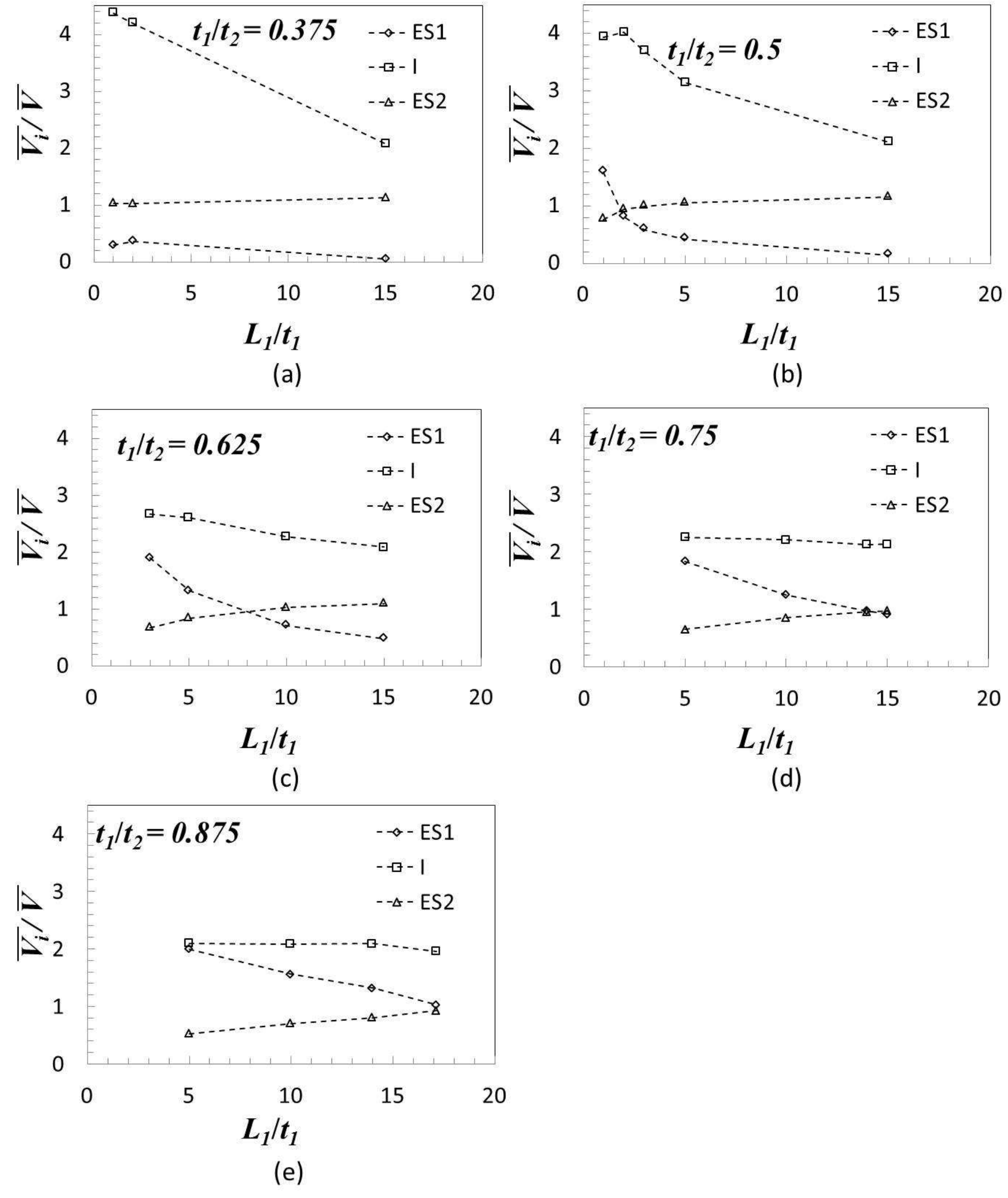


Figure 6.9: Effect of the length/thickness ratio of ES1 (L_1/t_1) on the normalized average velocity $\left(\frac{\overline{V_i}}{\overline{V}}\right)$ for die-T geometry with: (a) $t_1/t_2 = 0.375$; (b) $t_1/t_2 = 0.5$; (c) $t_1/t_2 = 0.625$; (d) $t_1/t_2 = 0.75$; (e) $t_1/t_2 = 0.875$.

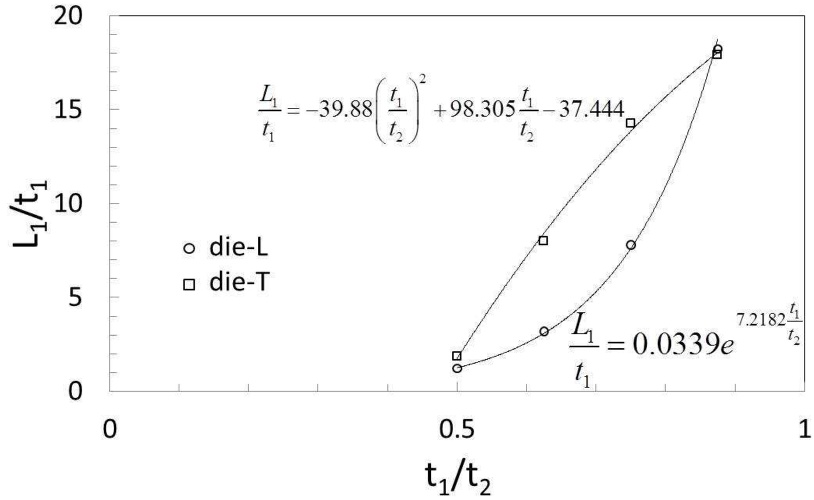
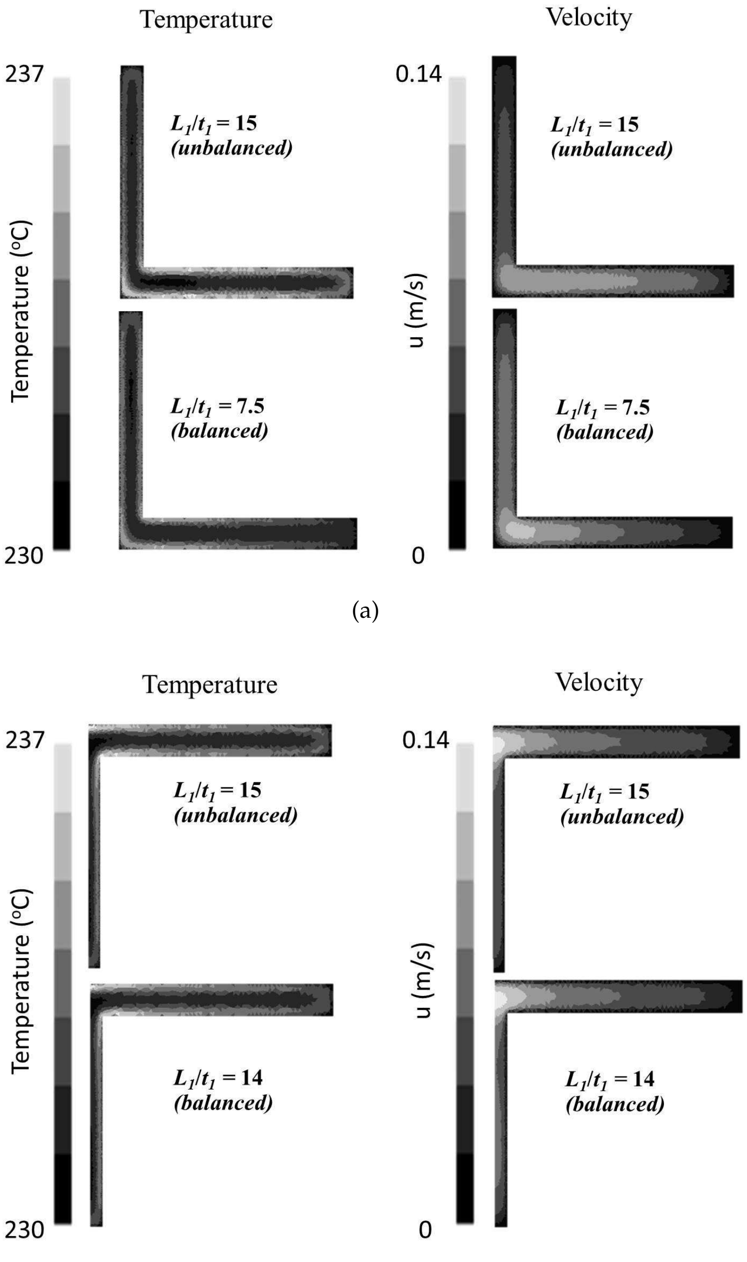


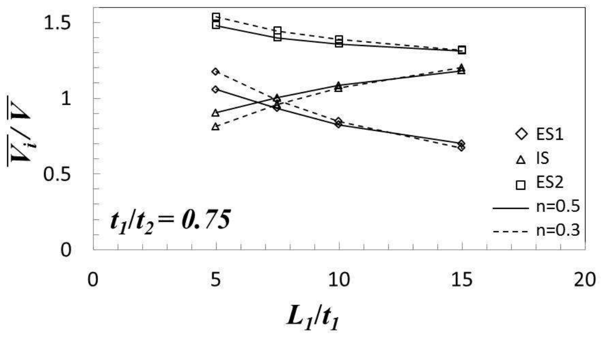
Figure 6.10: Variation of the intersection/balance point (between the ES1 and ES2 velocity data, L_1/t_1, with the ratio thickness, t_1/t_2.



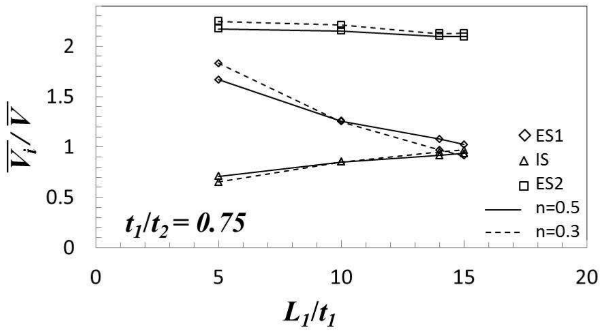
(a)

(b)

Figure 6.11: Velocity and temperature field contours, for the balanced and unbalanced flow for die-L and die-T geometries, with $t_1/t_2 = 0.75$: (a) die-L; (b) die-T.



(a)



(b)

Figure 6.12: Influence of the Power Law index on the flow distribution for a constant value of $t_1/t_2 = 0.75$. (a) die-L (b) die-T.

7

COOLING STAGE IN PROFILE EXTRUSION: AN IMPLICIT NUMERICAL APPROACH

Abstract[1]

In this work, a novel approach to model the cooling stage in profile extrusion, namely the heat exchange between two domains, is presented. For this purpose the discretization of the energy equation diffusive term, at the polymer-metal interface, is treated implicitly, thus avoiding the numerical oscillations and instabilities promoted by the usual iterative explicit approach. In this way, the two domains that comprise the system (Metal+Polymer) are treated as one, and the computations become more stable. To verify this implicit approach, the numerical results are compared with the analytical solutions obtained for the heat exchange between two simple slabs, and with numerical results obtained for more complex 2D geometries. Then, the developed code is tested on a 3D complex problem, involving the cooling stage of a wood plastic composite profile, for decking applications. The advantages of the implicit implementation are quantified for all the tested case studies.

7.1 INTRODUCTION

An extrusion line for the production of thermoplastics profiles usually comprises an extruder, a die, a calibration/cooling system, a haul-off unit and a saw (Fig. 7.1).

In the extruder, the polymer pellets are heated, melted and forced to advance. At the extruder exit, the molten material is progressively shaped by the extrusion die, which transforms the cylindrical flow, at the extruder outlet, into the desired

[1] N. D. Gonçalves, L. L. Ferrás, O. S. Carneiro, J. M. Nóbrega (2013). Cooling Stage in Profile Extrusion: an Implicit Numerical Approach, in preparation for submission in International Journal of Heat and Mass Transfer

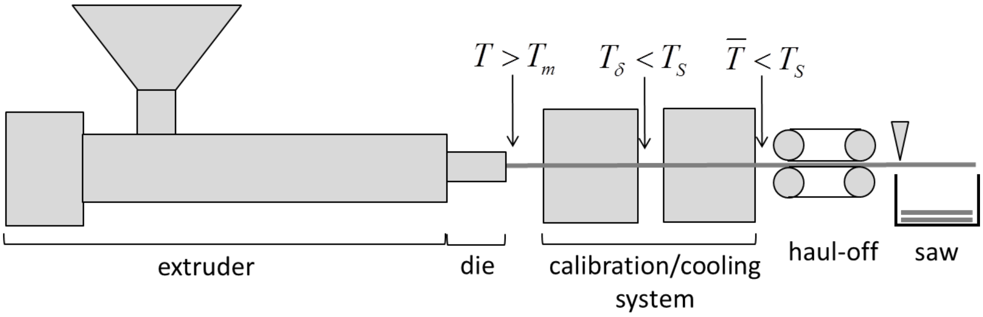


Figure 7.1: Schematic view of the extrusion and cooling processes.

cross-section geometry. After emerging from the die outlet section, the extrudate has an average temperature higher than its melt temperature, T_m, and, therefore, must be cooled down in order to maintain its shape. If several calibration/cooling units are used, in the first stage of calibration/cooling it suffices to lower the temperature, in a layer of thickness δ, to T_δ , a temperature lower than the solidification one, T_S. After the last calibration/cooling unit the profile average temperature should be lower than T_S, in order to avoid remelting (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004).

The aim of the numerical code developed is to predict the heat exchange between two domains comprising different materials, metal of the calibrator and polymeric extrudate, where the former is fixed in the space and the later moves at a pre-established velocity, U (Fig. 7.2). At the interface between calibrator and polymer profile, two boundary conditions can be considered: perfect contact, assuming temperature and heat flux continuity, and contact resistance, with heat flux continuity but with a temperature discontinuity, i. e., the temperature of the contacting faces are different (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004). At the outside walls of calibrator and polymer profile, adiabatic or natural convection can be considered. In Fig. 7.2 are summarized the boundary conditions that are usually considered in a typical cooling problem.

Polymer melts have a viscoelastic nature, which, combined with the instabilities inherent to the extrusion process and inhomogeneous material properties, turn out difficult the production of extrudates with the specified cross-section geometry and dimensions. Furthermore, as the extrudate advances along the extrusion line it is subjected to several external forces that may promote important deformations (Klein-

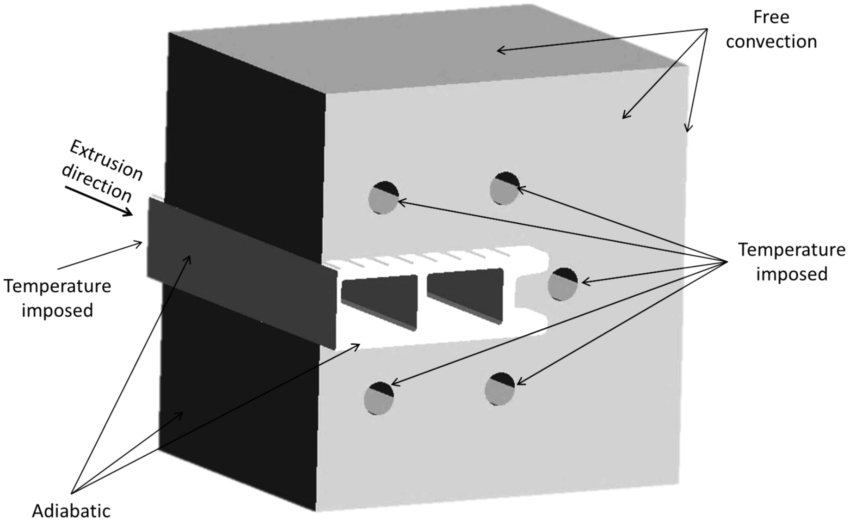


Figure 7.2: Example of calibrator (static) and profile (moving at a constant velocity)

dienst, 1973; Michaeli, 1992) if the cooling process does not ensure its adequate strength, which is achieved by the temperature reduction and consequent solidification. Thus, the performance of the calibration/cooling stage is crucial, since it defines the final dimensions of the extruded profile and guarantees enough stiffness to support the mechanical loads applied along the process (Kleindienst, 1973). In addition, the calibration/cooling system should assure the minimization of the temperature gradients developed in the profile cross-section during the cooling stage, in order to diminish the level of residual thermal stresses induced (Fradette et al., 1995; Brown and Enterprises, 2000).

Having in mind the above and the fact that there are many parameters affecting the thermal performance of the calibration/cooling system (e.g., extrusion conditions, system geometry, cooling conditions, vacuum conditions, polymer thermo-physical properties, properties of the calibrator material and profile cross-section) its design would benefit from the support of numerical tools, especially if an adequate performance is to be attained in a short time and at a reasonable cost.

In the available literature there are few works published concerning the development or the use of modelling tools for calibration/cooling system design purposes. The works of Kurz (Kurz, 1988) and Menges et al. (Menges et al., 1982) consist of 1D models and, therefore, were only applicable to ideal conditions, like uniform cooling and uniform thickness extrudates. Menges et al. (Menges et al., 1987) disregarded axial heat fluxes for a 2D modeling approach that was able to deal with any extrudate cross-section. Sheehy et al. (Sheehy et al., 1994) considered the axial diffusion through the so-called Corrected Slice Model (CSM), a hybrid 2D model that can handle with 3D problems. Szarvazy et al. (Szarvasy, 2000; Szarvasy and Sander, 1999) included more realistic boundary conditions, namely the radiation heat exchange within the internal cavities of hollow profiles in 2D. Pittman and Farah (Pittman and Farah, 1996) performed the prediction of sag flow in thick wall pipes, in 2D modelling studies of extrudate cooling. Other works related with this topic include that performed by Reifschneider et al. (Reifschneider et al., 2004) who used a transient finite element based model to study the cooling performance and shaping behavior of a calibrator, concluding that the calibrator heat transfer coefficient values are dependent upon the calibrator geometry and also of the extrusion flow rate. The relevance of the heat transfer coefficient was put in evidence in Nóbrega et al. (Nóbrega, Pinho, Oliveira and Carneiro, 2004), being, therefore, essential to characterize its value in conditions similar to those being modelled. Having this objective in mind, Mousseau et al. (Mousseau et al., 2009) presented a detailed description of an experimental apparatus in order to qualify and quantify the heat transfer modes. The results of the experimental work performed, together with the corresponding 3D numerical simulations done with the Fluent software, allowed the development of adequate thermal models.

Recently, Nóbrega et al. (Nóbrega, Pinho, Oliveira and Carneiro, 2004; Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004; Nóbrega and Carneiro, 2006; Nóbrega et al., 2008) developed and validated an algorithm that concerns the thermal design of calibrators for thermoplastic extrudates, comprising a non-isothermal 3D code based on the finite volume method (FVM), which modelled the thermal interchanges that take place during the cooling stage of profile extrusion, geometry and mesh generators and also an optimization routine aiming to determine the optimal cooling conditions. This work was performed with a numerical code that cope just with structured meshes, and it cannot be extended to more complex geometries due to the inherent limitations of those meshes. Ren et al. (Ren et al., 2010) also presented an optimization method for designing heterogeneous calibrator cooling systems but, instead, they used FEM and the finite difference method (FDM) to model the thermal interchanges.

As mentioned before, numerical tools play an important role in the study and optimization of heat transfer problems. When the physical domain is irregular and complicated, the use of FVM schemes is advantageous, since arbitrary volumes can be more easily used to discretize the physical domain (Dalal et al., 2008). In the FVM scheme two mesh types can be used: structured and unstructured (Ferziger and Peric, 2002; Blazek, 2006; Versteeg and Malalasekera, 2007). In structured meshes all the cells have the same number of neighbors, while in unstructured ones, each cell can have a different number of neighbors (Dalal et al., 2008), meaning that the latter have the capability of dealing with complex geometries (Sabbagh-Yazdi and Mastorakis, 2007; SabbaghYazdi and SaadatAbadi, 2011). In the literature it can be found several works that use FVM with unstructured meshes (Bašić et al., 2005; Bijelonja et al., 2006, 2005; Demirdžić et al., 2005; Demirdžić and Muzaferija, 1995; Teskeredžić et al., 2002), where several methods for the resolution of engineering problems are proposed. Some of the published works, concerned with heat transfer problems, using the FVM and unstructured meshes can be summarized as follows: Perron et al. (Perron et al., 2004), presented an approach where the solution variables are located at the circumcenter, and validated this new approach solving a forced 3D thermal flow in a cylinder; Wang and Joshi (Wang and Joshi, 2006) developed a computational tool, called Thermoflow, based on cell-centered collocated finite-volume method, that simulates the 3D incompressible viscous flow and heat transfer on unstructured meshes; Gonçalves (Gonçalves, 2007) developed a 3D numerical code based on FVM, with unstructured meshes, to solve problems of heat conduction in solid bodies and Araújo (Araújo, 2008) also developed a similar numerical 3D code to solve thermal and flow problems in the injection molding process.

Our research team is currently developing a methodology for the optimization of the design of calibrators, applicable to complex geometry profiles. For that, it is necessary to implement a new approach, capable of dealing with complex geometries, which comprises the development of a numerical code able to cope with unstructured meshes. The current version of the code solves the energy conservation equation in two domains (e.g. polymeric profile and metal calibrator), using an innovative implicit procedure for the implementation of the diffusive term, at the contact boundary, which is one of the main contributions of this work.

The remainder sections of this chapter are organized as follows. First, the numerical procedure and the discretization process will be presented, then, the code is verified, using both analytical solutions and numerical solutions obtained from a previously verified CFD code (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004). Finally, the robustness of the numerical code will be assessed using a real life case study of the cooling stage of a thermoplastic matrix composite deck profile. The chapter ends with the main conclusions obtained.

7.2 NUMERICAL PROCEDURE

The heat transfer that takes place within the calibration/cooling system is governed by the energy conservation equation that can be written as

$$\frac{\partial}{\partial t}\left(\rho_1 c_1 T\right) + \frac{\partial}{\partial x_i}\left(\rho_1 c_1 u_i T\right) - \frac{\partial}{\partial x_i}\left(k_1 \frac{\partial T}{\partial x_i}\right) = 0 \qquad (7.1)$$

for the polymer (with subscript 1) and as

$$\frac{\partial}{\partial t}\left(\rho_2 c_2 T\right) - \frac{\partial}{\partial x_i}\left(k_2 \frac{\partial T}{\partial x_i}\right) = 0 \qquad (7.2)$$

for the calibrator (with subscript 2), where ρ is the density, t is the time, u_i is the i^{th} velocity component, k is the thermal conductivity and c is the specific heat capacity.

One of the main issues when dealing with the interaction between two different materials, is the interface. At that region, the conservation and continuity rules should be accomplished by the numerical method.

For this particular problem, in the energy conservation equation (Eq. 7.1 and Eq. 7.2), the only term that has contributions to the interface between the two bodies, is the diffusive term,

$$\frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) \tag{7.3}$$

since there is no advection between both domains. At the interface two different conditions may occur: perfect contact or contact resistance (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004).

The FVM was the chosen method for the numerical solution of this partial differential equation. Following the FVM procedure, the physical domain is partitioned into smaller elements, the control volumes (CVs), transforming a continuous mathematical problem into a discrete one. The Eq. 7.1 and Eq. 7.2 are integrated in space (each control volume) and time (each time step), discretized and the non-linear terms are linearized, leading to a linear system of algebraic equations that can be solved implicitly or by using an iterative procedure (Versteeg and Malalasekera, 2007).

Following the usual FVM procedure (Patankar and Spalding, 1972) the integration (with the help of the divergence theorem) of the diffusive term (Eq. 7.3) over a control volume (Ω) is given by

$$\int\limits_{\Omega}\frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right)dV = \int\limits_{\partial\Omega} k\frac{\partial T}{\partial x_i}n_i dA, \tag{7.4}$$

where dA is the element area of the control volume surface ($\partial\Omega$).

Considering a constant value of the integrated term over each face, the previous expression (right-hand side of Eq. 7.4), can be written as follows

$$\sum_f A_f k\frac{\partial T}{\partial x_i}n_i, \tag{7.5}$$

which represents the contribution of the diffusive term, where f stands for a face of the control volume and n is the unit normal vector pointing outwards the surface (see Fig. 7.4).

The boundary conditions considered for the Energy Conservation Equation in cooling problems are:

- imposed temperature ($T = T^0$) in polymer profile at calibration stage beginning and cooling channels;
- insulated $\left(\frac{\partial T}{\partial x_i} n_i = 0\right)$ in internal profile boundaries and symmetry planes;
- natural convection ($k\frac{\partial T}{\partial x_i} n_i = \alpha \left(T_\infty - T_f\right)$, where α is the convection heat transfer coefficient at face f, T_∞ is the environment (far from the surface) temperature and T_f the face temperature) in external boundaries of calibrator or profile;
- contact resistance between two domains ($k\frac{\partial T}{\partial x_i} n_i = h_f \left(T_{f_1} - T_{f_2}\right)$, where h_f is the heat transfer coefficient at face f, T_{f_1} and T_{f_2} are the temperature of the contact faces f_1 and f_2, respectively) in contact faces between polymer and calibrator;
- perfect contact ($T_{f_1} = T_{f_2}$, where T_{f_1} and T_{f_2} are the temperature of the contact faces f_1 and f_2, respectively) in contact faces between polymer and calibrator.

The problems considered in this work are steady, but the unsteady term was considered to allow relaxation in the solving process.

Only the diffusive term will be studied in detail here. The numerical treatment of the remaining terms is described in Gonçalves et al. (2013).

The determination of temperature distribution in a polymer profile cooling problem starts with the assembling of system of energy conservation equations, that comprises a equation per each CV of the domains. The system of equations (or systems if there are sets of equations independent of others) are solved using an iterative procedure. Hereafter, the iterations performed by the system solver will be referred as *inner iterations*. With the results obtained for the elements, the boundary temperatures are updated according to their boundary conditions. If convergence is not achieved, i. e., if the initial error of the system of energy equations is higher than a predefined value, the process continues to the next time step (next *outer iteration*) (Fig. 7.3).

For the perfect contact case, the temperature in the contact surface of control volume 1 (CV1), T_{f1}, equals the temperature in the contact surface of CV2, T_{f2} and, therefore, $T_{f1} = T_{f2} = T_f$ (see Fig. 7.4). Since the diffusion terms for the calibrator

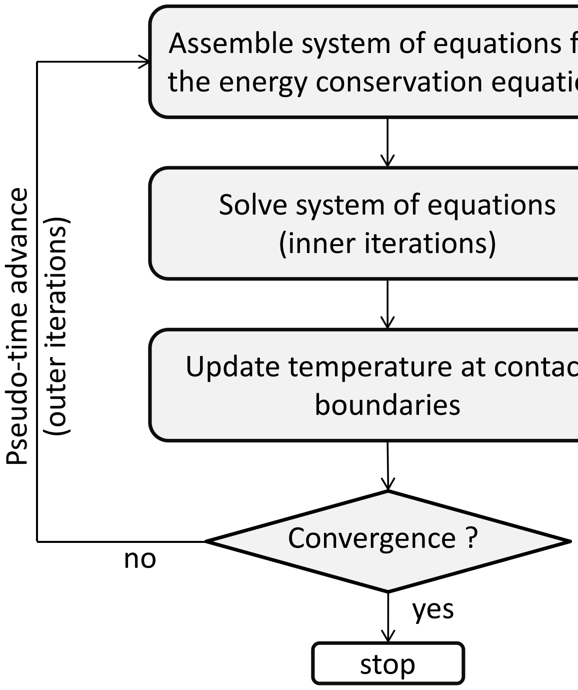

Assemble system of equations for the energy conservation equation
Solve system of equations (inner iterations)
Update temperature at contact boundaries
Convergence ?
no
yes
stop
Pseudo-time advance (outer iterations)

Figure 7.3: Iterative process implemented to solve the energy conservation equation.

and the polymer across the contact surface, are symmetric, in order to assure heat flux continuity we have that,

$$k_1 \left(\frac{\partial T}{\partial x_i} n_i \right)_1 = -k_2 \left(\frac{\partial T}{\partial x_i} n_i \right)_2 . \tag{7.6}$$

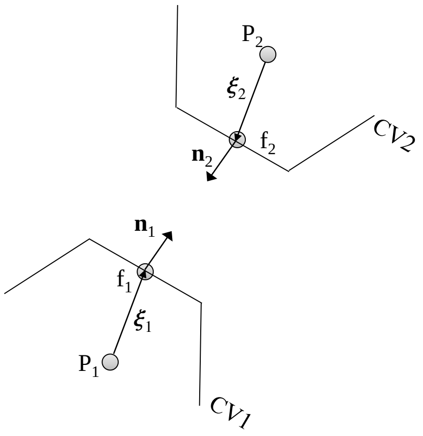


Figure 7.4: Typical arrangement of the computational domains at the interface boundary, between Control Volumes 1 (CV1) and 2 (CV2).

For the contact resistance case, we have that $T_{f1} \neq T_{f2}$ and heat flux continuity, being the diffusive term in face f_1 given by

$$A_f \left(k \frac{\partial T}{\partial x_i} n_i \right)_1 = A_f h \left(T_{f2} - T_{f1} \right) \tag{7.7}$$

where h is the interface heat transfer coefficient, T_{f1} and T_{f2} are the temperatures at CV_1 and CV_2 faces, respectively, and in face f_2 the diffusive term has a symmetric value of the one on the right-hand side of Eq. 7.7.

7.2.1 *Explicit approach*

The usual explicit procedure employed for this class of multi-domain problems (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004), divides the simulation domain into n smaller independent domains, where n is the number of bodies in contact, which results in n independent systems of equations. Thus the main problem to be solved is to find the correct temperature distribution at the contact face (interface boundary), for each decoupled domain (see Fig. 7.4) that accomplishes the selected boundary conditions. A possible strategy/procedure is to start with a guess boundary condition to Bodies 1 and 2, respectively BC1 and BC2 (Fig. 7.4), in order to assemble the system of equations for both domains. After solving these two systems of equations (performing inner iterations), the resulting temperature field in each domain is used to update the boundary values. The length of the vector residual of the system of equations $AT = b$, $\|r\| = \|AT - b\|$, is then evaluated and if its value is higher than a predefined one, the calculations proceed to the next outer iteration (pseudo-time advance) (Fig. 7.3). This explicit iterative procedure, would, in principle, converge to the correct solution, but the lack of control on the evolution for the temperature distribution may result in high instability, forcing the employment of high relaxation factors, which results in longer computational times. A possible solution to overcome these difficulties, is to implicit the interface boundary condition.

7.2.2 *Implicit approach*

Unlike the explicit approach, when the implicit approach is implemented, the contributions of the contact faces to their elements equations is done considering the current values of their CVs. This have two consequences: the temperatures at faces are evaluated with current values, instead of values obtained from last time step (as in the case of the explicit approach), and the face temperature can be written as function of values of both domains CVs, leading to a system of energy conservation equations that cannot be written as two independent systems.

Perfect contact

Assuming a one-sided approximation for the normal derivatives in Eq. 7.6, we obtain

$$k_1 \frac{T_f - T_{P1}}{(\xi_i n_i)_1} = k_2 \frac{T_{P2} - T_f}{(\xi_i n_i)_2} \tag{7.8}$$

where T_{Pi} represent the temperature variable at the centroid of the CV_i, ξ is a vector from CV centroid to boundary face centroid and n is the normal direction to each face (see Fig. 7.4). The previous expression can be further simplified, resulting in the following one, for T_f

$$T_f = \frac{cT_{P1} + T_{P2}}{c+1} \tag{7.9}$$

with $c = \frac{(\xi_i n_i)_2 k_1}{(\xi_i n_i)_1 k_2}$.

Replacing T_f in the expression of the contribution to the diffusive term in CV1 regarding the contact face (left-hand side of Eq. 7.10) by the right-hand side of Eq. 7.9, results in the right-hand side of Eq. 7.10. The similar contribution to CV2 is obtained with an analogous procedure using Eq. 7.11 and Eq. 7.9.

$$-A_f k_1 \frac{T_f - T_{P1}}{(\xi_i n_i)_1} = -\frac{A_f k_1}{(\xi_i n_i)_1} \left(\frac{c}{c+1} T_{P1} + \frac{1}{c+1} T_{P2} - T_{P1} \right) \tag{7.10}$$

$$-A_f k_2 \frac{T_f - T_{P2}}{(\xi_i n_i)_2} = -\frac{A_f k_2}{(\xi_i n_i)_2}\left(\frac{c}{c+1}T_{P1} - \frac{c}{c+1}T_{P2}\right) \tag{7.11}$$

We now have a unique set of algebraic equations that can be solved by an iterative procedure (performing only inner iterations). Note that since there are no explicit contributions in these terms, outer iterations are not required for this purpose. However, outer iterations are needed to update the estimates of inner faces terms. Nevertheless, the advantages of implicit approach are not affected.

Contact resistance

The temperature at the boundary faces can be approximated by the first two terms of a Taylor series expansion (assuming a smooth variation of the temperature field). This results in an expression for the interface temperatures T_{fi}, given by

$$T_{fi} = T_{Pi} + \overline{\left(\frac{\partial T}{\partial x_j}\right)_{Pi}}^{0} (\xi_k n_k)_i \, n_j \tag{7.12}$$

where the value of $\overline{\left(\frac{\partial T}{\partial x_j}e_j\right)_{Pi}}^{0}$ is evaluated using the least squares gradient reconstruction (Versteeg and Malalasekera, 2007) and the superscript 0 means that the gradient is calculated with the values from the previous (outer) iteration. Using the current value of element temperature $\left(T_{Pi}\right)$ instead of the previous one $\left(T^0_{Pi}\right)$ to evaluate the face temperature is the main difference of the implicit approach.

Replacing in right-hand side of Eq. 7.7 the value of T_{fi} by the correspondent right-hand side of Eq. 7.12, the original expression to the contribution of contact resistance face to the diffusive term of CV1 is obtained (Eq. 7.13).

$$A_f h \left(T_{f2} - T_{f1}\right) = A_f h \left(T_{P2} + \overline{\left(\frac{\partial T}{\partial x_j}\right)_{P2}}^{0} (\xi_k n_k)_2 \, n_j - T_{P1} - \overline{\left(\frac{\partial T}{\partial x_j}\right)_{P1}}^{0} (\xi_k n_k)_1 \, n_j\right) \tag{7.13}$$

The contribution to diffusive term of CV2 due the contact face is obtained in a similar fashion, and it is symmetric to the right-hand side of Eq. 7.13.

7.3 CASE STUDIES

In order to assess the proposed implicit implementation for the boundary interface, three distinct heat transfer case studies were considered, namely: i) two rectangular slabs (considering both perfect contact and contact resistance at the interface); ii) cooling of a polymer tape (considering perfect contact at the interface); iii) cooling of a complex geometry cross section profile (considering a contact resistance at the interface). The first two case studies allow to verify the code implementation, since the code predictions were compared with analytical or numerical results published in the literature, and the last one allows to test this new formulation in a complex and realistic problem. In the last case study, the code was also used to optimize the calibration/cooling system employed.

7.3.1 *Two rectangular slabs*

The first problem to be studied here, consists of two slabs ($S1$ and $S2$), with thermal conductivities k_1 and k_2, sharing a contact surface, as shown in Fig. 7.5 (a). For both perfect contact and contact resistance, the analytical solution was devised in Nóbrega, Carneiro, Covas, Pinho and Oliveira (2004), being given by:

$$T_1\left(x,y\right) = T_{b1} + \frac{2k_2\left(T_{b2}-T_{b1}\right)}{\pi\left(k_1+k_2\right)}\sum_{n=1}^{\infty}\frac{\left(-1\right)^{n+1}+1}{n}\frac{sin\left(\frac{n\pi x}{W}\right)sinh\left(\frac{n\pi y}{W}\right)}{sinh\left(\frac{n\pi H}{W}\right)} \tag{7.14}$$

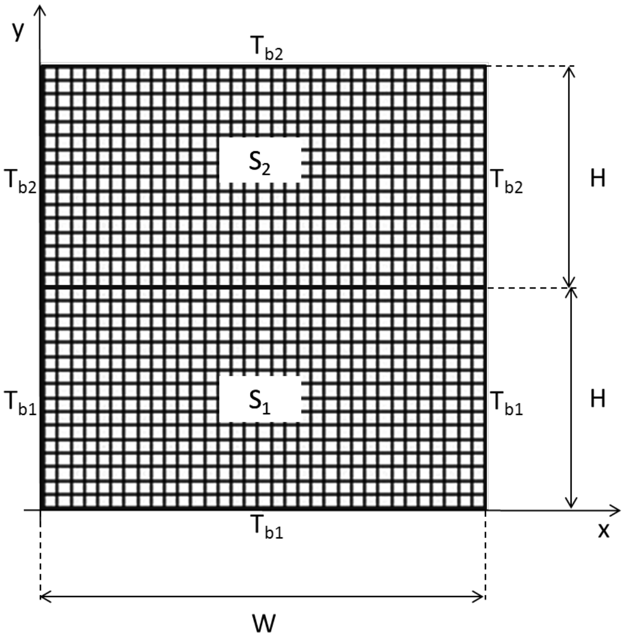
$$T_2\left(x,y\right) = T_{b2} + \frac{2k_1\left(T_{b1}-T_{b2}\right)}{\pi\left(k_1+k_2\right)}\sum_{n=1}^{\infty}\frac{\left(-1\right)^{n+1}+1}{n}\frac{sin\left(\frac{n\pi x}{W}\right)sinh\left(\frac{n\pi(-y+2H)}{W}\right)}{sinh\left(\frac{n\pi H}{W}\right)} \tag{7.15}$$

for the perfect contact case, and by

$$T_1\left(x,y\right) = T_{b1} + \frac{2h\left(T_{b2}-T_{b1}\right)}{\pi}\sum_{n=1}^{\infty}\frac{\left(-1\right)^{n+1}+1}{n}\frac{sin\left(\frac{n\pi x}{W}\right)sinh\left(\frac{n\pi y}{W}\right)}{k_1\frac{n\pi}{W}cosh\left(\frac{n\pi H}{W}\right)+h\frac{k_1+k_2}{k_2}sinh\left(\frac{n\pi H}{W}\right)} \tag{7.16}$$

$$T_2\left(x,y\right) = T_{b2} + \frac{2h\left(T_{b1}-T_{b2}\right)}{\pi}\sum_{n=1}^{\infty}\frac{\left(-1\right)^{n+1}+1}{n}\frac{sin\left(\frac{n\pi x}{W}\right)sinh\left(\frac{n\pi(-y+2H)}{W}\right)}{k_2\frac{n\pi}{W}cosh\left(\frac{n\pi H}{W}\right)+h\frac{k_1+k_2}{k_1}sinh\left(\frac{n\pi H}{W}\right)} \tag{7.17}$$

for the contact resistance case.



(a)

Figure 7.5: Two rectangular slabs case study: system geometry and mesh employed.

To verify the new implementation, the numerical predictions for the temperature distribution were compared with the analytical solutions. For that we used a mesh comprising 1056 elements (hexahedrons), having 33 and 32 cells along x and y, respectively (Fig. 7.5 (b)). The boundary conditions used were, $T_{b1} = 100\,^{o}C$ and $T_{b2} = 180\,^{o}C$ (Fig. 7.5). The thermal conductivities used were $k_1 = 7\,Wm^{-1}K^{-1}$ for $S1$ and $k_2 = 14\,Wm^{-1}K^{-1}$ for $S2$, and for the contact resistance case, the heat transfer coefficient used was $h = 500\,Wm^{-2}K^{-1}$.

The temperature fields, for perfect contact and contact resistance, are shown in Fig. 7.6, which evidences the difference between both interface boundary conditions, specially in what concerns to the temperature continuity at the interface, for the first, and discontinuity, for the last. In Fig. 7.7 we compare the analytical and numerical temperature distributions along the line $x = 0.5W$, which illustrates the very good agreement obtained.

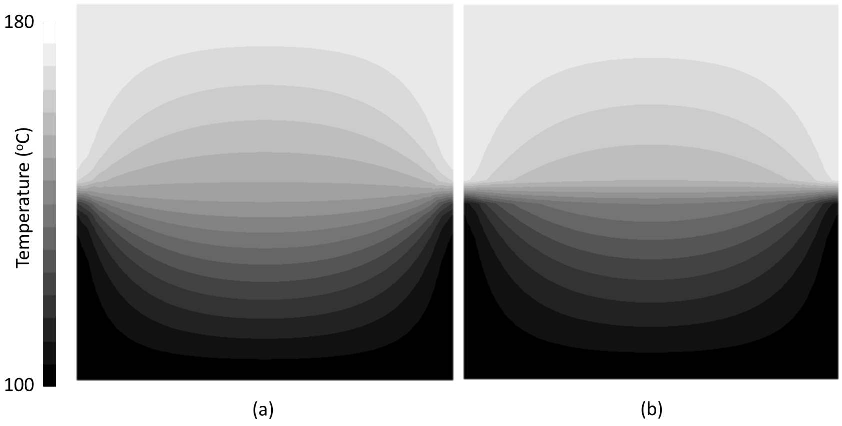
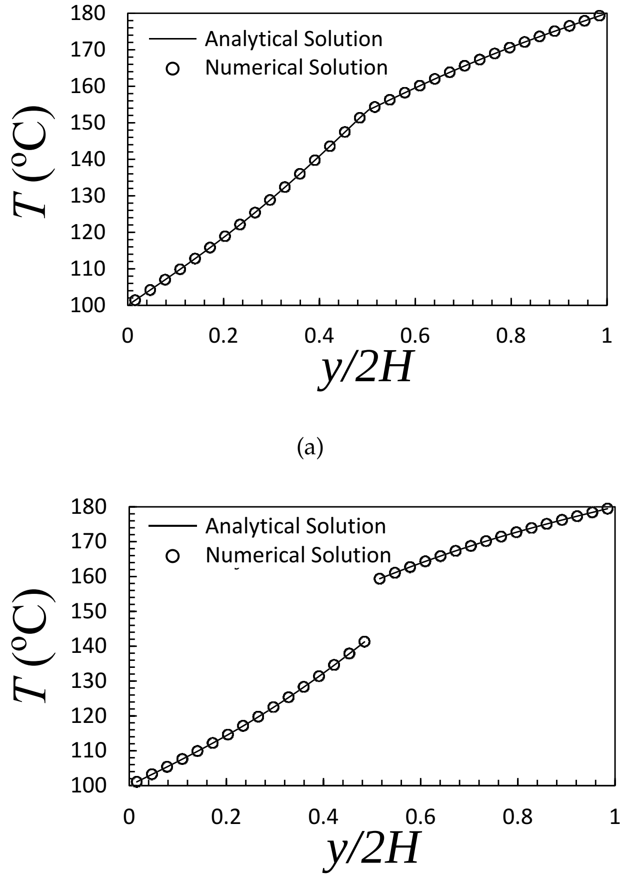


Figure 7.6: Temperature distribution for the two rectangular slabs case study: (a) perfect contact; (b) contact resistance.

In order to compare the implicit and explicit implementations, the two rectangular slabs problem was solved using these two case implementations with the mesh shown in Fig. 7.5. For comparison purposes the relative average error, between the analytical results and numerical predictions, of the temperature distribution along the line $x = 0.5W$ at each outer iteration was computed (Fig. 7.8).

Since the problems are stationary, the solution of the system of equations in each time step is not necessary, and the iterative solver performs just a limited number of iterations in each time step. When the stop criterion (to the iterative solver of the system of equations) is defined with a predefined decrease of the residual vector, two different situations occur: if the predefined decrease error is to small, the time step should be smaller to have convergence; if the predefined decrease error is higher, the number of iterations performed in non initial time steps become too higher, spending unnecessary time in intermediate time steps. Having this in mind, the stop criterion used in the iterative solver of the system of equations was a predefined number of iterations.

To define the number of iterations, it has to be considered that if it is too small, the changes performed in the solution estimate are small and the process become time



(a)

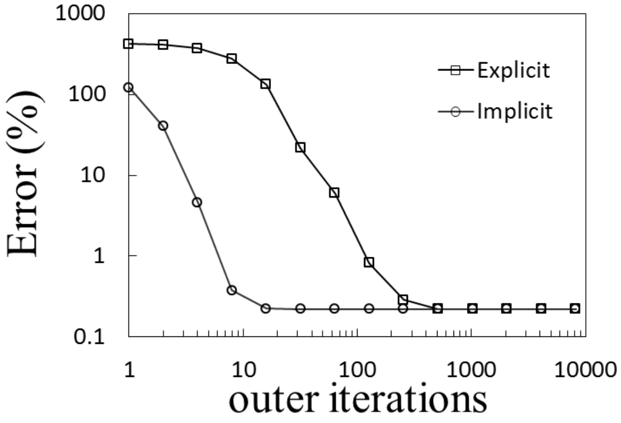
(b)

Figure 7.7: Comparison between the analytical and numerical temperature distributions, at x=0.5W, for the two rectangular slabs case study, assuming at the interface: (a) Perfect contact interface; (b) Contact resistance interface.

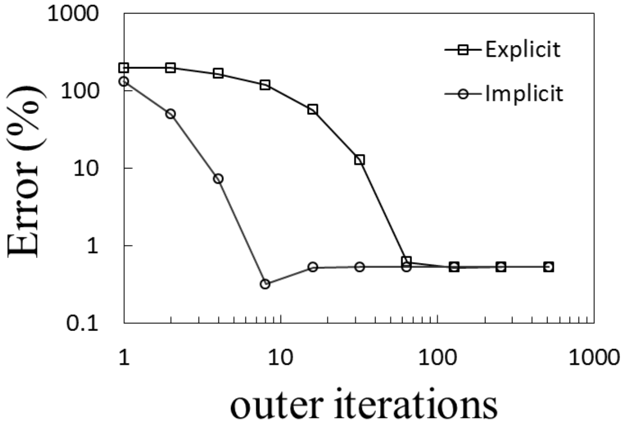
spending, on the other hand, if the number of iterations is too big, divergence can occur.

Perfect contact interface

Using the explicit and implicit schemes (with 100 inner iterations), we show in Fig. 7.8 (a) that the implicit approach reveals more efficiency, with a high degree of accuracy since the beginning of the calculation. As for the explicit approach, we can see that the relative error in the beginning of iterations is circa 420% for the explicit implementation and 122% for the implicit one. But as the number of outer iterations increases, the error also decreases and the numerical solution converges to the analytical solution. In the 16^{th} iteration, a relative error of 0.22% was found for the implicit approach, while, for the explicit one the same error magnitude was found only for the 512^{th} iteration.



(a)



(b)

Figure 7.8: Evolution of the difference between the numerical predictions and the analytical solution along the iterative process for the two rectangular slabs case study: (a) Perfect contact; (b) Contact resistance.

The implicit approach is more computationally demanding than the explicit one, but since it is more stable and evolutes faster to the solution, the increase in computational time is compensated. For example, the numerical simulation of this problem using the implicit approach requires circa 4% more calculation time than the one required for the explicit approach to run 1000 outer iterations, but it achieves the final solution in just 3% of the outer iteration required for the explicit approach. As overall balance, the ratio of the calculation time between the explicit and implicit approaches required to achieve the final solution is circa 32.

Contact resistance interface

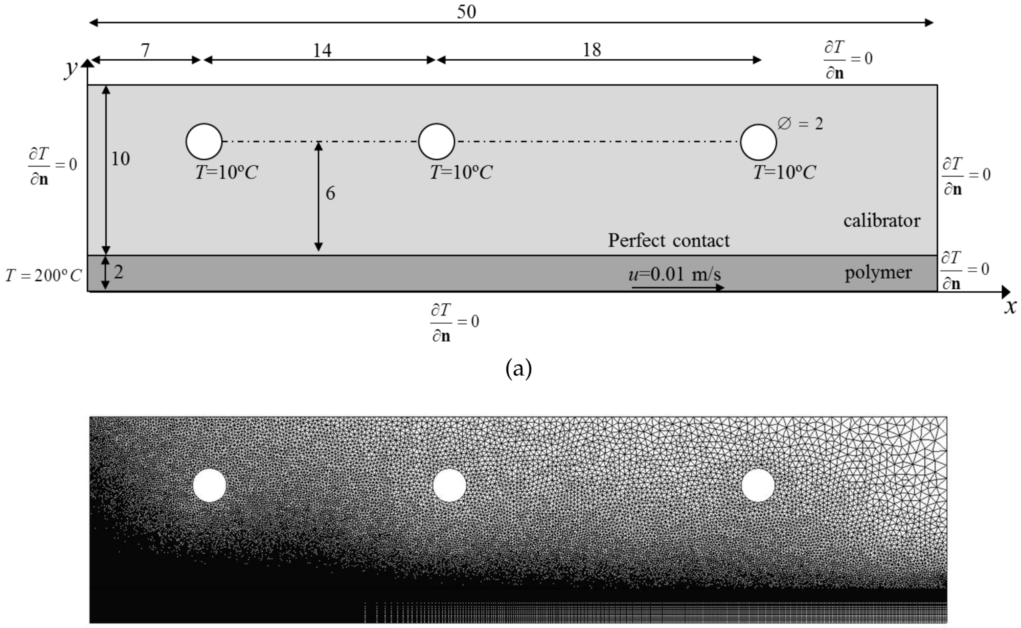
The number of inner iterations used to solve this problem was 100, because divergence occurred using the explicit approach with a higher number of inner iterations. This issue did not occurred for the implicit approach.

As shown in Fig. 7.8 (b), the decrease of the error is faster when the implicit approach is used. As the number of outer iterations increases both relative errors decrease, reaching error values of 0.53% for both the implicit and explicit approaches, but much earlier for the implicit approach. When using 1000 inner iterations the same error is achieved with only 4 outer iterations, which is an important issue since the use of faster system solvers allow to reduce the computation time. When the explicit approach is used, even with faster system solvers, a higher number of outer iterations must be used due divergence problems.

Unlike the perfect contact case, in the contact resistance case the implicit approach is less computational demanding than the explicit one since the increase of calculations needed to define the system coefficients is compensated with the fact that the flow between the contact faces is not estimated in an additional procedure. For this particular problem the numerical simulation using the implicit approach spent circa 8% less computational time than the explicit one to run 1000 outer iterations. Furthermore, the implicit approach only requires circa 25% of outer iterations to achieve the solution. As overall balance, the ratio of the calculation time between the explicit and implicit approaches required to achieve the final solution, is circa 4.3.

7.3.2 *Cooling of a polymer tape*

The second case study, consists of a metallic calibrator comprising three cooling channels (with an imposed temperature of $10\,^{o}C$) in contact with a moving (with velocity $u = 0.01\,m\,s^{-1}$) thin polymer tape, which is cooled as shown in Fig. 7.9 (a). The inlet temperature of the polymer tape is $200\,^{o}C$, and for the remaining external boundaries adiabatic conditions were imposed. At the interface between the calibrator and the polymer, perfect contact was assumed.


50
7
14
18
$\frac{\partial T}{\partial \mathbf{n}} = 0$
y
$\varnothing = 2$
$\frac{\partial T}{\partial \mathbf{n}} = 0$
10
T=10ºC
T=10ºC
T=10ºC
6
$\frac{\partial T}{\partial \mathbf{n}} = 0$
calibrator
Perfect contact
$T = 200°C$
2
u=0.01 m/s
polymer
$\frac{\partial T}{\partial \mathbf{n}} = 0$
x
$\frac{\partial T}{\partial \mathbf{n}} = 0$

(a)

(b)

Figure 7.9: Cooling of a polymer tape case study: (a) Geometry and boundary conditions; (b) Mesh used in the simulations. (dimensions in mm).

The calibrator (c subscript) and polymer (p subscript) properties are the following: thermal conductivities $k_p = 0.18\, W\, m^{-1} K^{-1}$ and $k_c = 23\, W\, m^{-1} K^{-1}$, density $\rho_p = 1400\, kg\, m^{-3}$ and specific heat capacity $c_p = 1000\, J\, kg^{-1} K^{-1}$. This problem has already been addressed in the literature by Nóbrega et al. (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004) to assess a numerical code based on the FVM, which work with structured meshes, developed to model the calibration stage of the extrusion process.

The 2D mesh used in the numerical simulations is illustrated in Fig. 7.9 (b). It comprises 178454 cells (101654 triangular prisms for the calibrator and 76800 hexahedrons for the polymer), with higher mesh refinement near the inlet face and along contact face, where higher temperature gradients are expected to occur.

The temperature distribution predicted by the numerical code, illustrated in Fig. 7.10, show that the temperature decreases when we move away from the inlet, for both calibrator and polymer melt. Note that the calibrator isothermal lines near the inlet are not perpendicular to the contact interface, as predicted by Sheehy et al.

(Sheehy et al., 1994), which was shown to be unrealistic by Nóbrega et al. (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004). Perpendicular isothermal lines would mean that no heat exchange could exist at the contact interface.

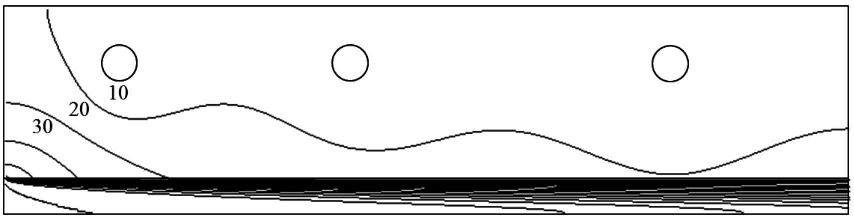


Figure 7.10: Cooling of a polymer tape case study: temperature isolines.

Nóbrega et al. (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004) computed the polymer average temperature at the outlet and obtained a value of $120.6\,^oC$. In this work, we obtained an average outlet temperature of 120.7 °C, in agreement with the previous result.

Our numerical results were also compared with the ones from Nóbrega et al. (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004). To do that, we extracted the temperature values at three different axial locations $x = 7\,mm$ (Fig. 7.11 (a)), $x = 30\,mm$ (Fig. 7.11 (b)) and $x = 50\,mm$ (Fig. 7.11 (c)) and plotted the three temperature profiles. Fig. 7.11 shows that a good agreement was obtained between the code predictions and the ones obtained by Nóbrega, Carneiro, Covas, Pinho and Oliveira (2004).

To solve this problem with the implicit numerical approach a time step of $10^9\,s$ and 1000 inner (system solver) iterations could be used. Due to lack of convergence, the employment of such a high time step was not possible if the explicit approach would be used. Therefore, in order to be able to compare both approaches, a time step of $1\,s$ was used. Using the implicit implementation circa 0.2% less computational time, when compared to the explicit implementation, is needed to run the 34988 outer iterations required to achieve the minimum error. But the ratio of the calculation time between the explicit and implicit approaches, required to achieve the final solution, is circa 3.5.

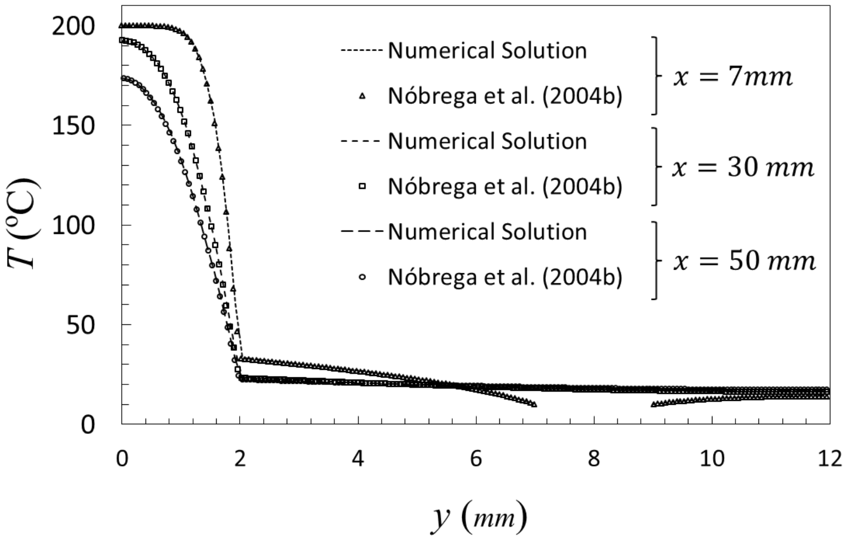


Figure 7.11: Cooling of a polymer tape case study: Comparison between the numerical results obtained for the calibrator-polymer simulations and the results provided by Nóbrega, Carneiro, Covas, Pinho and Oliveira (2004), for three different axial locations along the geometry: $x = 7\,mm$; $x = 30\,mm$; $x = 50\,mm$.

7.3.3 *Cooling of a complex geometry cross section profile*

The purpose of the third case study was to consider a real life application. For this sake, we performed heat exchange simulations between a polymer and a calibrator for the production of a wood plastic composite profile (Fig. 7.12 (a)). The calibrator comprises cooling channels with an imposed temperature of $20\,^oC$ and the polymer inlet temperature was set to $190\,^oC$. The polymer profile and calibrator dimensions are shown in Fig. 7.12 (a). For the cooling process, two calibrators in series were used, with a distance x between them, as shown in Fig. 7.12 (b). The effect of the distance between calibrators (x) will be studied in detail at the final part of this subsection.

The reasons for using two short calibrators, instead of a unique longer one, resides in the fact that the temperature distribution becomes more homogeneous with the existence of a gap between the two calibrators, and the efficiency of the cooling system increases (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004). This happens because when the polymer melt is inside the calibrator, the temperature at the contact surfaces (calibrator and polymer) become closer, and the heat exchange is lower. Therefore, when the polymer is in the space between calibrators, a more homogeneous

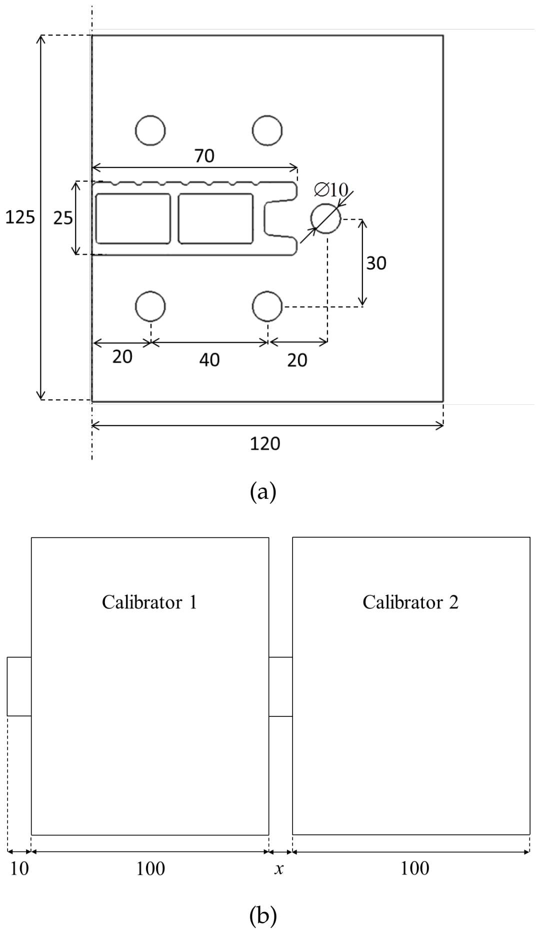


Figure 7.12: Cooling of a complex geometry cross section profile case study: (a) front view of the profile and calibrator; (b) side view of the calibrator/cooling system (composed by two calibrators) (dimensions in mm; due to symmetry reasons, only half of the geometry was considered in the simulations).

temperature is obtained (since the heat is conducted from the interior to the external boundaries) and when the polymer melt enters the subsequent calibrator, the heat exchange between the two domains is more effective, due to a higher temperature difference between them.

For this problem, we consider a contact resistance with a typical value of $h = 500\,W\,m^{-2}K^{-1}$ (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004) at the polymer-calibrator interface; for the external boundaries we assumed natural convection with $h = 5\,W\,m^{-2}K^{-1}$ and $T_{\infty} = 23\,^{o}C$ and the internal profile boundaries were considered insulated. The material properties considered are: thermal conductivities $k_p = 0.08\,W\,m^{-1}K^{-1}$ and $k_c = 20\,W\,m^{-1}K^{-1}$, density $\rho_p = 1200\,kg\,m^{-3}$ and specific heat capacity $c_p = 1300\,J\,kg^{-1}K^{-1}$. The polymer velocity was considered to be $0.2\,m\,min^{-1}$, a typical polymer extrusion velocity for this type of products.

A 3D overview and a cross section (perpendicular to the extrusion axis) of the mesh used in the numerical simulations can be seen in Fig. 7.13 (a), with 19538 triangles (15714 for the calibrator and 3825 for the polymer). The mesh consists of layers of triangular prisms along the extrusion axis, with 10 layers before the first calibrator, 20 layers in each calibrator, and a number of layers between calibrators depending on x, in order to maintain a similar mesh refinement level (Fig. 7.13 (b)). This number of elements was defined according to a mesh study.

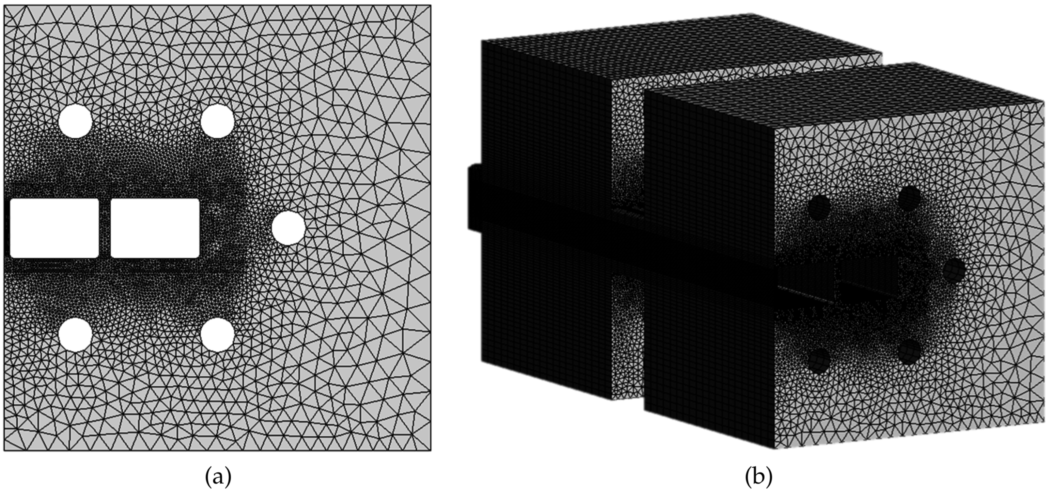


Figure 7.13: Cooling of a complex geometry cross section profile case study: (a) cross section mesh used in the simulations; (b) 3D mesh used in the simulations (due to symmetry reasons, only half of the geometry was considered in the simulations).

The temperature field distribution (at the outlet of the calibrating/cooling system) both for the polymer and the calibrator can be seen in Fig. 7.14, considering a gap between calibrators of $x = 600\,mm$.

In Fig. 7.14 one can see that the polymer surfaces in contact with the calibrator are at a lower temperature, but the inner of the polymer profile is still hot, due to the low thermal diffusivity of the polymer. Based on these results, some improvements could be performed in the calibration/cooling system geometry; for example, one could change the gap x , add an extra calibrator or adjust the cooling channel layout. Careful should be taken in this procedure, because the temperature reduction should

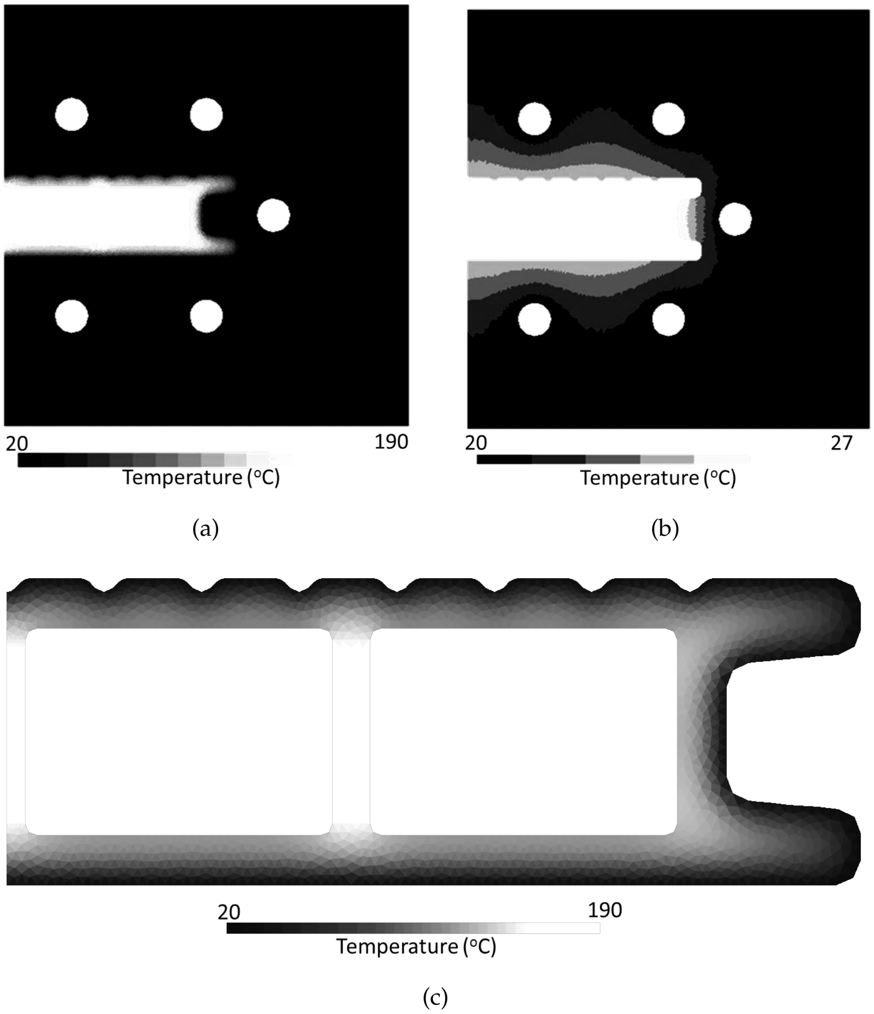


Figure 7.14: Cooling of a complex geometry cross section profile case study: polymer melt and calibrator temperature distribution at the outlet of the cooling system: (a) polymer and 2^{nd} calibrator; (b) 2^{nd} calibrator 2; (c) polymer.

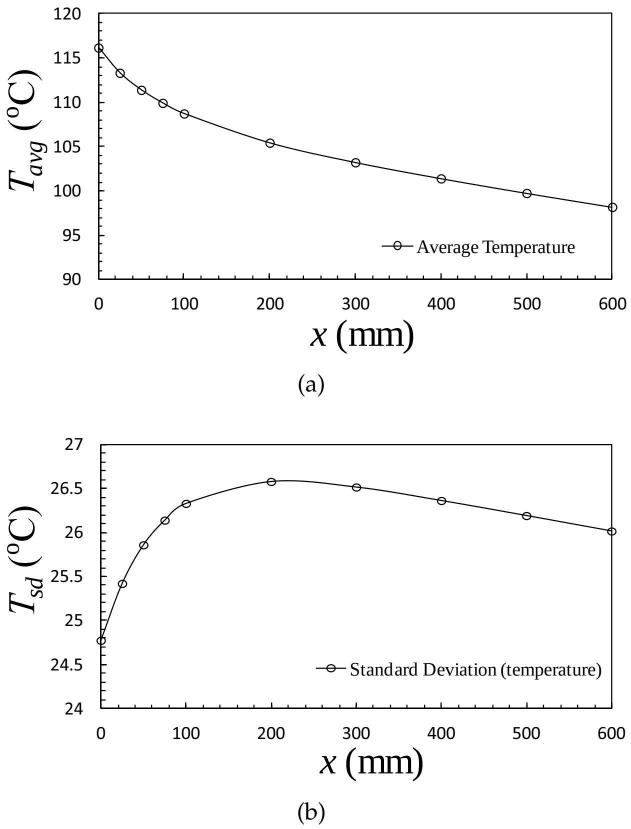
be as uniform as possible, in order to reduce the thermal induced residual stresses (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004). Keeping this in mind, a possible solution could be the use of cooling air in the interior of the mentioned profile, in order to be also able to reduce the temperature in the internal profile boundary.

Aiming to understand the system behavior, the variation of the outlet temperature as a function of the gap x was studied. For that, we plotted the average temperature, T_{avr} (Eq. 7.18), and the temperature standard deviation, T_{std} (Eq. 7.19), as function of the distance x between the two calibrators.

$$T_{avr} = \frac{\sum\limits_{f} A_f T_f}{\sum\limits_{f} A_f} \tag{7.18}$$

$$T_{std} = \sqrt{\frac{\sum\limits_{f} A_f \left(T_f - T_{avr}\right)^2}{\sum\limits_{f} A_f}} \tag{7.19}$$

As expected, the average temperature decreases as we increase x (Fig. 7.15 (a)). An interesting fact is that the standard deviation increases with x, for values of up to $200mm$, and then starts to decrease. This reveals the existence of a non-monotonic behavior of the temperature homogeneity as a function of the distance between the calibrators that should be taken into account when designing a calibration/cooling system.



(a)

(b)

Figure 7.15: Case study 3: average temperature and temperature standard deviation (at the exit of the second calibrator) as a function of the gap x.

Another possible way to reduce T_{avr} is to reduce the cooling fluid temperature (more energy consuming). Using one calibrator with a lower temperature, of $10\,^{o}C$, at the cooling channels we obtained a T_{avr} of $111.8\,^{o}C$ and a T_{std} of $26.24\,^{o}C$. Based on the results plotted in Fig. 7.15 a T_{avr} of $111.8\,^{o}C$ could also be achieved with two calibrators using a cooling temperature of $20\,^{o}C$ and a distance of $44\,mm$ between the two cooling units. For this case, the standard deviation was even smaller, $\mathrm{T}_{std} = 25.76\,^{o}C$, which emphasizes the advantage of splitting the cooling length into more than one unit, being this procedure also less energy consuming.

To compare the performance of the explicit and implicit implementations for this case study, 1000 inner iterations were used, and a time step of $1\,s$ was selected, since higher values lead to divergence, when the explicit approach is used, and lower values caused lower convergence rates, for both implementations. Using the implicit implementation, circa 6% less computational time is needed to run the 3540 outer iterations required to achieve the minimum error. The ratio of the calculation time between the explicit and implicit approaches, required to achieve the final solution, is circa 1.06.

7.4 CONCLUSIONS

The calibration/cooling stage of polymeric profiles consists mainly of a heat exchange process between the calibrator and the polymer. The design of those systems can benefit from the employment of appropriate numerical modeling tools. In this work a novel implicit implementation for the interface polymer-calibrator boundary condition was proposed and compared with the usually employed explicit approach. The implicit scheme is more stable and presents higher convergence rates than the explicit approach. In the case of perfect contact interface, this improvement proved to be much more relevant; however even with contact resistance interface the implicit approach allowed the achievement of lower calculations times.

Three case studies were analyzed, namely, the heat exchange between two slabs, a two dimensional heat transfer between a moving polymer tape and a calibrator, and, a real life complex profile cooling process.

In the case of the two slabs problem, the numerical code was verified comparing the analytical solution with the numerical results computed using both the explicit and the implicit approaches. The implicit approach improvement was more relevant in the perfect contact case, being circa 32 times faster, whereas for the contact resistance case it was circa 4.3 times faster than the explicit approach.

The second case study allowed to compare the new results with those of the work of Nóbrega et al. (Nóbrega, Carneiro, Covas, Pinho and Oliveira, 2004). In this case, the implicit approach was circa 3.5 times faster than the explicit one.

The last case study showed a real life application, involving the simulation of the cooling of a profile with complex cross section geometry. Such study evaluated the influence of the distance between two calibrators, showing that the temperature standard at the system outlet presents a non-monotonic behavior with the distance between calibrators. It was also shown that the employment of independent cooling units is a better alternative to the reduction of the cooling units temperature. The reduction of calculation time was not so relevant in this problem, being the problem solved 1.06 times faster when the implicit approach was used.

8

CONCLUSIONS

During this work, a numerical code that employs the Finite Volume Method on unstructured meshes to solve the mass, momentum and energy conservation equations, following a SIMPLE based algorithm, was developed. This numerical code allows to model the non-isothermal flow of generalized Newtonian fluids in confined channels of complex geometry.

The developed code was verified using typical benchmark problems, and, subsequently, was used to model the flow in extrusion dies involving complex geometry cross section profiles. The results obtained show that the developed code is useful to study and improve the extrusion die geometry.

The numerical code was also employed to aid the design of an extrusion die to produce a profile with complex cross-section geometry, a wood plastic composite (WPC) decking profile, and the results were compared with experimental results, revealing a good agreement. For this case, the initial trial geometry was improved through several iterations, changing the specific flow channel dimensions. This procedure led to good results, since a better flow distribution was achieved.

A parallelized version of the numerical code, implemented on GPU, was compared with its serial counterpart, showing speedups between 3.5x and 7x. The studies performed allowed to conclude that the speedup increases with the mesh size, thus higher speedups are obtained for more refined meshes, the most computational demanding ones. These advantages were obtained without any special memory management on the GPU, which leaves room for more improvements.

Also within the die design framework, two different extrusion dies, for the production of "L" and "T" shaped profiles, were studied, since they represent a base in which complex geometries can be decomposed. This study aimed the identification of the relation between the parallel zone length, as function of the elemental section thickness ratio, required to obtain a balanced flow distribution at the die outlet section. The results showed that for ratios of thicknesses between 0.5 and 0.875 it was possible to achieve a balanced flow at the outlet by changing the length of the pre-parallel zone, according to an exponential relation in the die-L case, and a quadratic relation, in the die-T case. The sensitivity to different values of the power-law index n, pointed out the need to perform a more comprehensive study.

In what concerns the calibration system design, an implicit formulation for the polymer-calibrator interface was implemented. It was shown that the implicit approach is faster than the explicit one, requiring less computational time and being also a more stable process, allowing for fast convergence. The problem used to verify this code, the heat exchange between two slabs, revealed a reduction of 97% of the computation effort for the case where perfect contact was assumed and of 75% when a contact resistance at the slabs interface was considered. To the WPC profile cooling case, the reduction of computational time was circa 6%, when the implicit implementation was employed instead of explicit one.

9

FUTURE WORK

The production of profiles by extrusion comprises several processes, such as the forming stage and calibration stage, that, nowadays, can be numerically simulated separately. The interdependence between these processes requires the development of numerical codes able to deal with the several production stages. During this work, a numerical code able to simulate separately the non-isothermal flow of generalized Newtonian fluids in extrusion dies and the cooling process that occurs on the calibration stage was developed. It would be useful to develope a numerical code that could be able to simulate these processes in an integrated way.

Presently, the outlet flow channel faces are treated as faces with null gradient on the normal direction, but for the case of free surface problems (such as the flow between the die and the calibrator), the imposed pressure would be the appropriate approach. It would be useful to implement this new boundary condition in order to model the polymer flow between the extrusion die and the calibrator, where the extrudate swell occurs. Another requirement to model the extrudate swell is to consider the viscoelasticity of the material, since elasticity effects play a major role in this stage. In this work it was employed the Generalized Newtonian fluids as an approach to viscoelastic fluids, but the improvement of the current numerical code in order to deal with viscoelastic fluids, would be useful to make the simulation more realistic to polymer extrusion cases.

The cooling stage is modelled using a parameter, the interface heat transfer coefficient, that is treated as constant. However, this parameter depends on several parameters and plays an important role on the heat exchanges at the interface. Thus, the characterization of this coefficient for real extrusion conditions and its dependence on process parameters, would be useful to model the heat exchange between the polymer and the calibrator with more realism.

The present numerical code can be used to aid the design of profile extrusion dies, but the analysis of the obtained results and the consequent changes on geometry are still committed to the designer. Algorithms to search the optimal geometry combined with the developed numerical code, would be useful to automatize the optimization process and make it more independent of the designer's experience.

REFERENCES

Alonso, J., Mitty, T., Martinelli, L. and Jameson, A. (1994). A two-dimensional multigrid-driven navier-stokes solver for multiprocessor architectures, *Proceedings of the Parallel CFD*, Vol. 94.

Amdahl, G. (1967). Validity of the single processor approach to achieving large scale computing capabilities, *Proceedings of the April 18-20, 1967, spring joint computer conference*, ACM, pp. 483–485.

ANSYS Polyflow (2013).
URL: *http://www.ansys.com*

Araújo, B. (2008). *Integrated three-dimensional flow and thermal simulation of the injection moulding process*, PhD thesis.

Asouti, V., Trompoukis, X., Kampolis, I. and Giannakoglou, K. (2010). Unsteady cfd computations using vertex-centered finite volumes for unstructured grids on graphics processing units, *International Journal for Numerical Methods in Fluids* **67**(2): 232–246.

Bašić, H., Demirdžić, I. and Muzaferija, S. (2005). Finite volume method for simulation of extrusion processes, *International journal for numerical methods in engineering* **62**(4): 475–494.

Bhardwaj, M., Day, D., Farhat, C., Lesoinne, M., Pierson, K. and Rixen, D. (2000). Application of the feti method to asci problems - scalability results on 1000 processors and discussion of highly heterogeneous problems, *International Journal for Numerical Methods in Engineering* **47**(1-3): 513–535.

Bharti, R. P., Chhabra, R. P. and Eswaran, V. (2006). Steady flow of power law fluids across a circular cylinder, *The Canadian Journal of Chemical Engineering* **84**: 406–421.

Bijelonja, I., Demirdžić, I. and Muzaferija, S. (2005). A finite volume method for large strain analysis of incompressible hyperelastic materials, *International journal for numerical methods in engineering* **64**(12): 1594–1609.

Bijelonja, I., Demirdžić, I. and Muzaferija, S. (2006). A finite volume method for incompressible linear elasticity, *Computer methods in applied mechanics and engineering* **195**(44): 6378–6390.

Blazek, J. (2001). *Computational fluid dynamics: principles and applications*, Elsevier Science.

Blazek, J. (2006). *Computational Fluid Dynamics: Principles and Applications*, second edn, Elsevier Science.

Bolz, J., Farmer, I., Grinspun, E. and Schröoder, P. (2003). Sparse matrix solvers on the gpu: conjugate gradients and multigrid, *ACM Transactions on Graphics (TOG)*, Vol. 22, ACM, pp. 917–924.

Brandvik, T. and Pullan, G. (2008). Acceleration of a 3d euler solver using commodity graphics hardware, *46th AIAA Aerospace Sciences Meeting*, Citeseer.

Brown, R. and Enterprises, F. (2000). Predicting how the cooling and resulting shrinkage of plastics affect the shape and straightness of extruded profiles, *SPE/ANTEC 2000 Proceedings*, CRC, p. 383.

Brown, R. J., Kim, H. T. and Summers, J. W. (1979). Pratical principles of die design, *SPE, 37th Annual Technical Conference*.

Carley, J. (1954). Flow of melts in"crosshead"-slit dies; criteria for die design, *Journal of applied physics* **25**(9): 1118–1123.

Carneiro, O. and Nóbrega, J. (2012). *Design of Extrusion Forming Tools*.

Carneiro, O., Nóbrega, J., Covas, J., Oliveira, P. and Pinho, F. (2004). A study on the thermal performance of calibrators, *ANTEC 2004, SPE Annual Technical Conference*, Vol. 1, pp. 251–255.

Carneiro, O., Nóbrega, J., Oliveira, P. and Pinho, F. (2003). Flow balancing in extrusion dies for thermoplastic profiles part ii: Influence of the design strategy, *International Polymer Processing* **18, 3**: 307.

Carneiro, O., Nóbrega, J., Pinho, F. and Oliveira, P. (2001). Computer aided rheological design of extrusion dies for profiles, *Journal of Materials Processing Technology* **114**(1): 75–86.

Carneiro, O., Nóbrega, J., Pinho, F. and Oliveira, P. (2003). Automatic design of profile extrusion dies: Experimental assessment, *PPS 2003, Polymer Processing Society Meeting*, pp. 1–2.

Carneiro, O. S., Nóbrega, J. M., Pinho, F. T. and Oliveira, P. J. (2004b). Automatic balancing of profile extrusion dies: Experimental assessment, *ANTEC 2004, SPE Annual Technical Conference*, Vol. 50, pp. 91–95.

Castro, M., Ortega, S., De la Asuncion, M., Mantas, J. and Gallardo, J. (2011). Gpu computing for shallow water flow simulation based on finite volume schemes, *Comptes Rendus Mécanique* **339**(2): 165–184.

Cebeci, T., Shao, J., Kafyeke, F. and Laurendeau, E. (2005). *Computational fluid dynamics for engineers: from panel to navier-stokes methods with computer programs*, Springer.

Chen, C., Jen, P. and Lai, F. (1997). Optimization of the coathanger manifold via computer-simulation and an orthogonal array method, *Polymer Engineering & Science* **37**(1): 188–196.

Chesshire, G. and Jameson, A. (1989). Flo 87 on the ipsc/2- a parallel multigrid solver for the euler equations, *Applications of supercomputers in engineering: Fluid flow and stress analysis applications* pp. 79–88.

Chung, C. and Lohkamp, D. (1976). Designing coat-hanger dies by power-law approximation, *Modern Plastics* **53**(3): 52.

Cohen, J. and Molemaker, M. (2009). A fast double precision cfd code using cuda, *Parallel Computational Fluid Dynamics: Recent Advances and Future Directions* pp. 414–429.

Corrigan, A., Camelli, F., Löhner, R. and Wallin, J. (2009). Running unstructured grid based cfd solvers on modern graphics hardware, *19th AIAA Computational Fluid Dynamics Conference*, pp. 1–11.

Dalal, A., Eswaran, V. and Biswas, G. (2008). A finite-volume method for navier-stokes equations on unstructured meshes, *Numerical Heat Transfer, Part B: Fundamentals* **54**(3): 238–259.

Darwish, M. and Moukalled, F. (2003). Tvd schemes for unstructured grids, *International Journal of heat and mass transfer* **46**(4): 599–611.

Dellagiacoma, F., Vitaletti, M., Jameson, A., Martinelli, L., Sibilla, S. and Visintini, L. (1995). Flo67p: a multi-block version of flo67 running within paragrid, *Parallel Computational Fluid Dynamics, Elsevier Science Publishers BV (North–Holland), Amsterdam* pp. 199–206.

Demirdžić, I., Džaferović, E. and Ivanković, A. (2005). Finite-volume approach to thermoviscoelasticity, *Numerical Heat Transfer, Part B* **47**(3): 213–237.

Demirdžić, I. and Muzaferija, S. (1995). Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology, *Computer methods in applied mechanics and engineering* **125**(1): 235–255.

Dieflow (2013).
URL: *http://www.dieflow.com/*

Dongarra, J., Foster, I., Fox, G., Gropp, W., Kennedy, K., Torczon, L. and White, A. (2003). *Sourcebook of parallel computing*, Vol. 3003, Morgan Kaufmann Publishers San Francisco, CA.

Dostál, Z. (2009). *Optimal quadratic programming algorithms: with applications to variational inequalities*, Vol. 23, Springer.

Dostál, Z., Gomes Neto, F. and Santos, S. (2000). Solution of contact problems by feti domain decomposition with natural coarse space projections, *Computer methods in applied mechanics and engineering* **190**(13): 1611–1627.

Dostál, Z., Horák, D. and Vlach, O. (2007). Feti-based algorithms for modelling of fibrous composite materials with debonding, *Mathematics and Computers in Simulation* **76**(1): 57–64.

Dostál, Z., Neto, F. and Santos, S. (2000). Duality-based domain decomposition with natural coarse-space for variational inequalities, *Journal of Computational and Applied Mathematics* **126**(1): 397–415.

Elgeti, S., Probst, M., Windeck, C., Behr, M., Michaeli, W. and Hopmann, C. (2012). Numerical shape optimization as an approach to extrusion die design, *Finite Elements in Analysis and Design* **61**: 35–43.

Elsen, E., LeGresley, P. and Darve, E. (2008). Large calculation of the flow over a hypersonic vehicle using a gpu, *Journal of Computational Physics* **227**(24): 10148–10161.

Endrass, B. (1993). High-speed extrusion of profiles, *Kunststoffe German Plastics(Germany)* **83**(8): 584–586.

Ettinger, H., Sienz, J., Pittman, J. and Polynkin, A. (2004). Parameterization and optimization strategies for the automated design of upvc profile extrusion dies, *Structural and Multidisciplinary Optimization* **28**(2): 180–194.

Farhat, C., Chen, P., Mandel, J. and Roux, F. (1998). The two-level feti method part ii: Extension to shell problems, parallel implementation and performance results, *Computer methods in applied mechanics and engineering* **155**(1): 153–179.

Farhat, C., Mandel, J. and Roux, F. (1994). Optimal convergence properties of the feti domain decomposition method, *Computer methods in applied mechanics and engineering* **115**(3): 365–385.

Fatahalian, K., Sugerman, J. and Hanrahan, P. (2004). Understanding the efficiency of gpu algorithms for matrix-matrix multiplication, *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, ACM, pp. 133–137.

Ferziger, J. H. and Peric, M. (2002). *Computatinal Methods for Fluid Dynamics*, 3rd edn, Springer.

FLOW 2000 (2013).
URL: *http://www.futuresoft.net/F2k-ext.htm*

Fox, G., Williams, R. and Messina, P. (1994). *Parallel computing works!*, Morgan Kaufmann Pub.

Fradette, L. and Tanguy, P. (1996). On the determination of heat transfer coefficient between pvc and steel in vacuum extrusion calibrators, *International Journal of Numerical Methods for Heat & Fluid Flow* **6**(1).

Fradette, L., Tanguy, P., Thibault, F., Sheehy, P., Blouin, D. and Hurez, P. (1995). Optimal design in profile extrusion calibration, *Journal of Polymer Engineering* **14**(4): 295–322.

Frey, P. and George, P. (2010). *Mesh generation*, Wiley-ISTE.

Ghia, U., Ghia, K. N. and Shin, C. T. (1982). High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method, *Journal of Computational Physics* **48**: 387–411.

Glowinski, R. (1997). *Domain decomposition methods in sciences and engineering: 8th international conference, Bejing, PR China*, John Wiley & Son Ltd.

Gonçalves, N. D. (2007). *Finite volume method on unstructured meshes (in portuguese)*, Master's thesis, Faculdade de Ciências da Universidade do Porto.

Gonçalves, N. D., Carneiro, O. S. and Nóbrega, J. (2013). Design of complex profile extrusion dies through numerical modeling, *Journal of Non-Newtonian Fluid Mechanics* **200**: 103–110.

Goodnight, N., Woolley, C., Lewin, G., Luebke, D. and Humphreys, G. (2005). A multigrid solver for boundary value problems using programmable graphics hardware, p. 193.

Gordon, W., Van Tyne, C. and Moon, Y. (2007). Overview of adaptable die design for extrusions, *Journal of materials processing technology* **187**: 662–667.

Hagen, T., Lie, K. and Natvig, J. (2006). Solving the euler equations on graphics processing units, *Computational Science–ICCS 2006* pp. 220–227.

Hall, J., Carr, N., Hart, J. et al. (2003). Cache and bandwidth aware matrix multiplication on the gpu.

Hoffmann, K. and Chiang, S. (2000). {Computational fluid dynamics, Vol. 1}, *Wichita, KS: Engineering Education System* .

Hristov, V. and Vlachopoulos, J. (2007). Thermoplastic silicone elastomer lubricant in extrusion of polypropylene wood flour composites, *Advances in Polymer Technology* **26**: 100–108.

Hurez, P., Tanguy, P. and Blouin, D. (1996). A new design procedure for profile extrusion dies, *Polymer Engineering & Science* **36**(5): 626–635.

HyperXtrude (2013).
URL: *http://www.altairhyperworks.com/Product,15,HyperXtrude.aspx*

J. Dooley, G. Oliver, K. X. (2009). Breakthrough inventions and innovations in die design for polymer processing, *Annual Technical Conference - ANTEC, Conference Proceedings*, Vol. 2, pp. 684–690.

Jost, G., Jin, H., an Mey, D. and Hatay, F. F. (2003). *Comparing the OpenMP, MPI, and hybrid programming paradigms on an SMP cluster*, Vol. 3.

Kampolis, I., Trompoukis, X., Asouti, V. and Giannakoglou, K. (2010). Cfd-based analysis and two-level aerodynamic optimization on graphics processing units, *Computer Methods in Applied Mechanics and Engineering* **199**(9): 712–722.

Kirk, D. B. and Wen-mei, W. H. (2010). *Programming massively parallel processors: a hands-on approach*, Morgan Kaufmann.

Kleindienst, V. (1973). Number 63(1), Kunststoffe.

Kostic, M. and Reifschneider, L. (2006). Design of extrusion dies, *Encyclopedia of Chemical Processing DOI* **10**.

Koziey, B., Vlachopoulos, J., Vlcek, J. and Svabik, J. (1996). Profile die design by pressure balancing and cross flow minimization, *Technical Papers of the Annual Technical Conference-Society of Plastics Engineers Incorporated*, SOCIETY OF PLASTICS ENGINEERS INC, pp. 247–252.

Krüger, J. and Westermann, R. (2003). Linear algebra operators for gpu implementation of numerical algorithms, *ACM Transactions on Graphics (TOG)*, Vol. 22, ACM, pp. 908–916.

Kruis, J. (2004). Domain decomposition methods on parallel computers, *Progress in Engineering Computational Technology* pp. 299–322.

Kruis, J. (2007). *Domain decomposition methods for distributed computing*, Saxe-Coburg Publications.

Kruis, J., Matouš, K. and Dostál, Z. (2002). Solving laminated plates by domain decomposition, *Advances in Engineering Software* **33**(7): 445–452.

Kurz, H. (1988). Fixing the dimensions of extruded high-mw polyolefin pipes during cooling, *Kunstst. Ger. Plast.* **78**(11): 1052–1058.

Lazou, C. (1988). Supercomputers and their use.

Lebaal, N., Schmidt, F. and Puissant, S. (2009). Design and optimization of three-dimensional extrusion dies, using constraint optimization algorithm, *Finite Elements in Analysis and Design* **45**(5): 333–340.

Legat, V. and Marchal, J. (1993). Die design: an implicit formulation for the inverse problem, *International journal for numerical methods in fluids* **16**(1): 29–42.

Legat, V. and Marchal, J.-M. (1992). Three-dimensional extrusion: an implicit formulation for gn flows, *Theorectical and Applied Rheology, Proc. XIth Int. Congr. on Rheology*, Elsevier Science Publishers B. V.

Lehnhäuser, T. and Schäfer, M. (2005). A numerical approach for shape optimization of fluid flow domains, *Computer methods in applied mechanics and engineering* **194**(50): 5221–5241.

Liu, P., Xie, S. and Cheng, L. (2012). Die structure optimization for a large, multicavity aluminum profile using numerical simulation and experiments, *Materials & Design* **36**: 152–160.

Liu, Y., Liu, X. and Wu, Enhuf title = Real-time 3D fluid simulation on GPU with complex obstacles, y. . . p. . . b. . C. o. . I. o. . n. t. . . (n.d.).

Magoules, F. (2008). *Mesh partitioning techniques and domain decomposition methods*, Saxe-Coburg Publications.

Magoules, F. (2010). *Substructuring techniques and domain decomposition methods*, Saxe-Coburg Publications.

Marchal, J. M. and Goublomme, A. (2000). Parametric optimization of extrusion dies through numerical simulation, *3rd Esaform Conference on Material Forming*.

Maria, A. (1997). Introduction to modeling and simulation, *Proceedings of the 29th conference on Winter simulation*, IEEE Computer Society, pp. 7–13.

Mattson, T., Sanders, B. and Massingill, B. (2005). *Patterns for parallel programming*, Addison-Wesley Professional.

McKelvey, J. and Ito, K. (1971). Uniformity of flow from sheeting dies, *Polymer Engineering & Science* **11**(3): 258–263.

Menges, G., Haberstroh, E. and Janke, W. (1982). Systematic lay-out of cooling lines for film, sheet and pipe extrusion plant, *Kunststoffe-german plastics* **72**(6): 332–336.

Menges, G., Kalwa, M. and Schmidt, J. (1987). Finite element simulation of heat transfer in plastics processing, *Kunstst. Ger. Plast.* **77**(8): 797–802.

Michaeli, W. (1992). *Extrusion Dies for Plastic and Rubber: Design and Engineering Computations*, 2nd ed spe books edn, Hanser Publishers.

Michaeli, W. (2003). *Extrusion dies for plastics and rubber: design and engineering computations*, Hanser Gardner Publications.

Michaeli, W. and Kaul, S. (2004). Approach of an automatic extrusion die optimization, *Journal of polymer engineering* **24**(1-3): 123–136.

Michaeli, W., Kaul, S. and Wolff, T. (2001). Computer-aided optimization of extrusion dies, *Journal of polymer engineering* **21**(2-3): 225–238.

Monakov, A., Lokhmotov, A. and Avetisyan, A. (2010). Automatically tuning sparse matrix-vector multiplication for gpu architectures, *High Performance Embedded Architectures and Compilers* pp. 111–125.

Mousseau, P., Delaunay, D. and Lefevre, N. (2009). Analysis of heat transfer in pvc profiles during the extrusion calibration/cooling step, *International Polymer Processing* **24**(2): 122–132.

MPI (2013).
URL: *http://www.open-mpi.org/*

Mu, Y., Zhao, G., Wu, X. and Zhang, C. (2010). An optimization strategy for die design in the low-density polyethylene annular extrusion process based on fes/bpnn/nsga-ii, *The International Journal of Advanced Manufacturing Technology* **50**(5): 517–532.

Na, S. and Lee, T. (1997). Shape optimization of polymer extrusion die by three-dimensional flow simulation, *High Performance Computing on the Information Superhighway, HPC Asia'97*, IEEE, pp. 601–604.

Nóbrega, J. M. (2004). *Computer Aided Design of Forming Tools for the Production of Extruded Profiles*, PhD thesis, University of Minho.

Nóbrega, J. M. and Carneiro, O. S. (2006). Optimising cooling performance of calibrators for extruded profiles, *Plastics, rubber and composites* **35**(9): 387–392.

Nóbrega, J. M., Carneiro, O. S., Covas, J. A., Pinho, F. T. and Oliveira, P. J. (2004). Design of calibrators for extruded profiles. part i: Modeling the thermal interchanges, *Polymer Engineering & Science* **44**(12): 2216–2228.

Nóbrega, J. M., Carneiro, O. S., Gaspar-Cunha, A. and Gonçalves, N. (2008). Design of calibrators for profile extrusion - optimizing multi-step systems, *International Polymer Processing* **23**(3): 331–338.

Nóbrega, J. M., Carneiro, O. S., Pinho, F. T. and Oliveira, P. J. (2000a). Equilibrio do escoamento em cabeças de extrusÃ£o para perfis, *Segundo Encontro Nacional da Sociedade Portuguesa de Reologia*, pp. 217–225.

Nóbrega, J. M., Carneiro, O. S., Pinho, F. T. and Oliveira, P. J. (2000b). On the automatic die design for extrusion of thermoplastic profiles, *16th Annual Meeting of the Polymer Processing Society, PPS 16*.

Nóbrega, J. M., Carneiro, O. S., Pinho, F. T. and Oliveira, P. J. (2003). Flow balancing in extrusion dies for thermoplastic profiles - part i: Automatic design, *International Polymer Processing* **18**(3): 298–306.

Nóbrega, J. M., Carneiro, O. S., Pinho, F. T. and Oliveira, P. J. (2004). Flow balancing in extrusion dies for thermoplastic profiles - part iii: Experimental assessment, *International Polymer Processing* **19**: 225–235.

Nóbrega, J. M., Carneiro, O. S., Pinho, F. T., Oliveira, P. J. and Covas, J. A. (2003). Thermal aspects of the calibration of extruded profiles, 19th Int. Annual Meeting of the Polymer Processing Society (PPS-19), Melbourne, Australia.

Nóbrega, J. M., Pinho, F. T., Oliveira, P. J. and Carneiro, O. S. (2004). Accounting for temperature-dependent properties in viscoelastic duct flows, *International journal of heat and mass transfer* **47**(6): 1141–1158.

NVIDIA (2013). Cuda home page, www.nvidia.com.

Ohshima, S., Kise, K., Katagiri, T. and Yuba, T. (2007). Parallel processing of matrix multiplication in a cpu and gpu heterogeneous environment, *High Performance Computing for Computational Science-VECPAR 2006* pp. 305–318.

OpenMP (2013).
URL: *http://www.openmp.org*

Owens, J., Houston, M., Luebke, D., Green, S., Stone, J. and Phillips, J. (2008). Gpu computing, *Proceedings of the IEEE* **96**(5): 879–899.

Panda, S. K. and Chhabra, R. P. (2010). Laminar flow of power-law fluids past a rotating cylinder, *Journal of Non-Newtonian Fluid Mechanics* **165**: 1442–1461.

Patankar, S. and Spalding, D. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *International Journal of Heat and Mass Transfer* **15**(10): 1787–1806.

Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation.

Pearson, J. (1964). Non-newtonian flow and die design, *Trans. J. Plast. Inst* **32**: 239–44.

Pereira, S. (2011). *Parallelization of flow modelling code on a gpu*, Master's thesis, Universidade do Minho.

Pereira, S., Vuik, K., Pinho, F. and J.M., N. (2013). On the performance of a 2d unstructured cfd code on a gpu, *to be submitted* .

Perron, S., Boivin, S. and Hérard, J. (2004). A new finite volume discretization scheme to solve 3d incompressible thermal flows on unstructured meshes, *International journal of thermal sciences* **43**(9): 833–848.

Pittman, J. and Farah, I. (1996). Comprehensive simulation of cooling process in plastic pipe manufacture, *Plastics rubber and composites processing and applications* **25**(6): 305–312.

Pittman, J., Whitham, G., Beech, S. and Gwynn, D. (1994). Cooling and wall thickness uniformity in plastic pipe manufacture, *International Polymer Processing* **9**(2): 130–140.

Placek, L., Svabik, J. and Vlcek, J. (2000). Cooling of extruded plastic profiles, *Proceedings of the Annual Technical Conference of the Society of Plastics Engineers*.

Proctor, B. (1972). Flow analysis in extrusion dies, *SPE Journal* **28**(2): 34–41.

PROFILECAD (2013).
URL: *http://www.polydynamics.com/profile.htm*

Reifschneider, L., Kostic, M. and Vaddiraju, S. (2004). Computational design of u-profile die and calibrator, *ANTEC-CONFERENCE PROCEEDINGS-.*, Vol. 1, pp. 246–250.

Reilly, E. (2003). *Milestones in computer science and information technology*, Greenwood Publishing Group.

Ren, L., Yang, R. and Zhang, W. (2010). Optimization design in cooling system of heterogeneous calibrator for plastic profile extrusion, *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*, Vol. 2, IEEE, pp. 468–472.

Reuther, J., Rimlinger, M., Alonso, J. and Jameson, A. (1996). Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers, *AIAA, NASA, and ISSMO, Symposium on Multidisciplinary Analysis and Optimization, 6 th, Bellevue, WA*.

Sabbagh-Yazdi, S. and Mastorakis, E. (2007). Efficient symmetric boundary condition for galerkin finite volume solution of 3d temperature field on tetrahedral meshes, *WSEAS International Conference on Heat Transfer, Thermal Engineering and Environment (HTE'07), Athens, Greece*.

SabbaghYazdi, S. and SaadatAbadi, T. (2011). Sequential computations of two-dimensional temperature profiles and thermal stresses on an unstructured triangular mesh by gfvm method, *International Journal of Civil Engineering* **9**(3).

Schiedrum, H. O. (1983). *Kunststoffe German Plastics(Germany)* **73**(1): 2.

Schut, J. H. (2003). Computer flow analysis helps develop new profile dies faster, *Plastic Technology* .

Shahreza, A. R., Behravesh, A. H., Jooybari, M. B. and Soury, E. (2010). Design , optimization , and manufacturing of a multiple-thickness profile extrusion die with a cross flow, *Polymer Engineering & Science* **50**: 2417–2424.

Sheehy, P., Tanguy, P. and Blouin, D. (1994). A finite element model for complex profile calibration, *Polymer Engineering & Science* **34**(8): 650–656.

Sienz, J., Bates, S. and Pittman, J. (2006). Flow restrictor design for extrusion slit dies for a range of materials: Simulation and comparison of optimization techniques, *Finite elements in analysis and design* **42**(5): 430–453.

Sienz, J., Goublomme, A. and Luege, M. (2010). Sensitivity analysis for the design of profile extrusion dies, *Computers & Structures* **88**(9–10): 610–624.

Smith, D. (2003). Design sensitivity analysis and optimization for polymer sheet extrusion and mold filling processes, *International journal for numerical methods in engineering* **57**(10): 1381–1411.

Soury, E., Behravesh, A., Nasrabadi, H. and Zolfaghari, A. (2009). Design and manufacture of an extrusion die for wood-plastic composite, *Journal of Reinforced Plastics and Composites* **28**(12): 1433–1439.

Sterling, T., Becker, D., Salmon, J. and Savarese, D. (1999). *How to build a Beowulf: A Guide to the Implementation and Application of PC Clusters*, MIT press.

Stone, H. (1992). *High-performance computer architecture*, Addison-Wesley Longman Publishing Co., Inc.

Sweby, P. (1984). High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM journal on numerical analysis* **21**(5): 995–1011.

Szarvasy, I. (2000). Simulation of complex pvc window profile cooling during calibration with particular focus on internal heat exchange, *3rd ESAFORM Conference on Material Forming*.

Szarvasy, I. and Sander, R. (1999). Tailored design of extrusion dies, *Kunststoffe- Plast Europe* **89**(6): 7–9.

Szarvasy, I., Sienz, J., Pittman, J. and Hinton, E. (2000). Computer aided optimisation of profile extrusion dies-definition and assessment of the objective function, *International Polymer Processing* **15**(1): 28–39.

Tadmor, Z., Broyer, E. and Gutfinger, C. (1974). Flow analysis network (fan)-a method for solving flow problems in polymer processing, *Polymer Engineering & Science* **14**(9): 660–665.

Tadmor, Z. and Cogos, C. (2006). *Principles of Polymer Processing*, 2nd edition edn, John Wiley and Sons Inc.

Teskeredžić, A., Demirdžić, I. and Muzaferija, S. (2002). Numerical method for heat transfer, fluid flow, and stress analysis in phase-change problems, *Numerical Heat Transfer: Part B: Fundamentals* **42**(5): 437–459.

Tezduyar, T., Aliabadi, S., Behr, M., Johnson, A. and Mittal, S. (1993). Parallel finite-element computation of 3d flows, *Computer* **26**(10): 27–36.

Thompson, J. F., Soni, B. K. and Weatherill, N. P. (1999). *Handbook of Grid Generations*, CRC press.

Topping, B. (2000). *Parallel and distributed processing for computational mechanics: systems and tools*, Civil-Comp press.

Topping, B. (2002). *Computational Mechanics Using High Performance Computing*, Saxe-Coburg Publications.

Topping, B. and Ivanyi, P. (2010). *Parallel, Distributed and Grid Computing for Engineering*, Saxe-Coburg Publications.

Topping, B. and Khan, A. (1996). *Parallel finite element computations*, Vol. 10, Saxe-Coburg Publications.

Tropsa, V. (2001). *Predicting residual stresses due to solidification in cast plastic plates*, PhD thesis, University of London (United Kingdom).

Ulysse, P. (1999). Optimal extrusion die design to achieve flow balance, *International journal of machine tools and manufacture* **39**(7): 1047–1064.

Ulysse, P. (2002). Extrusion die design for flow balance using fe and optimization methods, *International journal of mechanical sciences* **44**(2): 319–341.

Versteeg, H. K. and Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics - The Finite Volume Method*, second edn, PEARSON.

Walshaw, C. and Cross, M. (2008). Jostle-multilevel graph partitioning software: An overview, *Mesh Partitioning Techniques and Domain Decomposition Methods* p. 27.

Walters, K. and Webster, M. (2003). The distinctive cfd challenges of computational rheology, *International journal for numerical methods in fluids* **43**(5): 577–596.

Wang, Q. and Joshi, Y. (2006). Algebraic multigrid preconditioned krylov subspace methods for fluid flow and heat transfer on unstructured meshes, *Numerical Heat Transfer, Part B: Fundamentals* **49**(3): 197–221.

Wördenweber, B. (1984). Finite element mesh generation, *Computer-Aided Design* **16**(5): 285–291.

Wortberg, J., Haberstroh, E., Lutterbeck, J., Masberg, U., Schmidt, J. and Targiel, G. (1982). Designing of extrusion lines, *Advances in Polymer Technology* **2**(2): 75–106.

Yu, Y. and Liu, T. (1998). A simple numerical approach for the optimal design of an extrusion die, *Journal of Polymer Research* **5**(1): 1–7.