Universidade do Minho
Escola de Engenharia

Ricardo Daniel Pereira da Costa

Advanced computational methods towards high-performance polymer processing simulations

December 2020

Advanced computational methods towards high-performance polymer processing simulations

Ricardo Daniel Pereira da Costa

UMinho | 2020

Universidade do Minho
Escola de Engenharia

Ricardo Daniel Pereira da Costa

Advanced computational methods towards high-performance polymer processing simulations

Doctoral thesis in
Science and Engineering of Polymers and Composites

Work accomplished under the supervision of
Professor João Miguel de Amorim Novais da Costa Nóbrega
Professor Stéphane Louis Clain

December 2020

Direitos de autor e condições de utilização do trabalho por terceiros

Acknowledgements

I acknowledge the financial support by IMPULSE – Polymers and Composites: Drivers of Technological Innovation and Industrial Competitiveness (project no. NORTE-08-5369-FSE-000034), co-funded by the North Portugal Regional Operational Programme (NORTE 2020), through the Portugal 2020 and the European Social Fund (FSE). I also acknowledge the computing facilities support by Search-ON2: Revitalization of HPC Infrastructure of UMinho (project no. NORTE-07-0162-FEDER-000086), co-funded by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF).

Statement of integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, 10th December 2020

Name: Ricardo Daniel Pereira da Costa

Signature: ____

Resumo

A aplicação da modelação computacional em problemas de engenharia no processamento de polímeros assistiu a um crescimento notável nos últimos anos, permitindo às indústrias utilizar ferramentas poderosas de conceção assistida por computador. Além do desenvolvimento de computadores potentes para superar as limitações computacionais, o desenvolvimento de métodos numéricos proficientes e precisos também contribuiu significativamente para a aplicação da modelação computacional em problemas de engenharia outrora intratáveis. No entanto, problemas cada vez mais complexos no processamento de polímeros devido às geometrias comumente elaboradas, aos processos não-isotérmicos e fluidos poliméricos com comportamento reológico complexo, incrementam claramente a necessidade de maior precisão numérica e eficiência computacional.

O principal objetivo do presente trabalho prende-se com a melhoria do desempenho, em termos de precisão numérica e eficiência computacional, das ferramentas computacionais empregues na resolução de problemas complexos na área do processamento de polímeros. Nesse sentido, métodos numéricos avançados são desenvolvidos no contexto do método dos volume finitos de forma a obter uma convergência do erro com o refinamento de malha maior do que as clássicas primeira e segunda ordens, desta forma resultando em ganhos substanciais de precisão. Para além disso, uma implementação eficiente dos métodos numéricos propostos é também desenvolvida, elaborando algoritmos que reduzem o custo computacional das simulações e, ao mesmo tempo, tirem também partido dos atuais processadores com capacidade de cálculo paralelo.

Uma análise e verificação aos desenvolvimentos numéricos e à implementação computacional foi exaustivamente levada a cabo com casos de estudo específicos para avaliar o desempenho dos métodos e algoritmos propostos. Os resultados obtidos comprovam que os métodos propostos atingem ordens de convergência elevadas e ótimas para o erro, sendo capazes de obter efetivamente, com malhas significativamente mais grosseiras, o mesmo nível de precisão da solução em comparação com os métodos de primeira e segunda ordens. Adicionalmente, ganhos substanciais em eficiência computacional, quer em termos de tempo de execução quer dos requisitos de memória, também são alcançados, dado que os algoritmos propostos potenciam essas melhorias sem perda de precisão numérica.

Os desenvolvimentos conseguidos representam um avanço importante para simulações mais precisas e computacionalmente mais eficientes de aplicações complexas no processamento de polímeros.

Palavras-chave: Modelação computacional, métodos numéricos, ordem de convergência elevada, processamento de polímeros, simulações de alta performance

Abstract

The application of the computational modelling in engineering problems of polymer processing has seen a remarkable growth in the past years, providing valuable computer-aided design tools to the related industries. Besides the development of powerful hardware to overcome the computational limitations, the development of proficient and accurate numerical methods has also significantly contributed to the applicability of the computational modelling to once intractable engineering problems. However, the ever-increasing complex polymer processing applications, comprising intricate three-dimensional geometries, non-isothermal processes, and polymeric fluids with complex rheological behaviour, clearly still raise the demand of numerical accuracy and computational efficiency.

The main objective of the present work is to improve the performance, either in terms of numerical accuracy and computational efficiency, of computational modelling tools to solve complex problems akin to polymer processing applications. In that regard, advanced numerical methods are developed in the finite volume method context, capable of providing an error convergence under mesh refinement higher than the classical first- and second-orders, therefore resulting in substantial accuracy gains. Moreover, the implementation efficiency of the proposed numerical methods is also addressed with algorithms that reduce the computational cost of the simulations, also taking advantage of modern multi-core processor computers.

A comprehensive analysis and verification, both of the numerical developments and the computational implementations, were exhaustively carried out with specific case studies to assess the performance of the proposed methods and algorithms. The obtained results prove that the proposed methods achieve the optimal high-order of convergence for the error and are capable of effectively obtaining the same solution accuracy level given by lower-order ones with significantly coarser meshes. Additionally, substantial gains in computational efficiency, both in terms of running time and memory usage, are also achieved, since the proposed algorithms further enhance these improvements without loss of numerical accuracy.

The achieved developments represent a significant advance towards more accurate and more computationally efficient simulations of complex polymer processing applications.

Keywords: Computational modelling, high-performance simulations, numerical methods, polymer processing, very high-order of convergence

Publications and awards

The following publications were prepared, submitted, and published in peer-reviewed international journals based on the present work.

Table of contents

List of figures

List of tables

List of symbols

Constants

g	Gravitational acceleration (9.8 m/s^2)

Greek symbols

κ	Thermal conductivity (W/m.K)
ρ	Density (kg/m^3)

Roman symbols

$\boldsymbol{u}$	Velocity vector (m/s^2)
C_p	Specific heat coefficient (J/kg.°C)
F	Heat flux (W/m^2)
Re	Reynolds number
T	Temperature (°C)
g	Gram
kg	Kilogram
m	Metre
m	Millimetre
N	Newton
s	Second
W	Watt

CHAPTER 1 Introduction

1.1 Polymer processing

The polymer processing industry emerged in 1868 [1] and is nowadays an essential society sector, delivering many everyday items indispensable for a variety of purposes. Polymeric materials, which include rubbers and plastics, have also stimulated major technological advance in many fields, for instance, health and transportation, among several others. These materials present complex microscopic structures, leading to counter-intuitive physical properties, sometimes deviating significantly from the expected behaviour, typical of other common materials. Consequently, the manufacturing technologies of polymeric materials have evolved into complex thermo-mechanical processes, and have been raising challenging problems for engineers, which require profound knowledge and experience. The present section further explores the science and engineering behind polymeric materials, some manufacturing technologies, and some current and emergent challenges.

1.1.1 Polymers

Polymers are long molecular chains consisting of repetitive subunit molecules, having considerably lower molecular mass, referred to as monomers. The chemical process of aggregating monomers together to form a polymer chain is referred to as polymerization, as exemplified in Figure 1.1 for the case of the polystyrene, a polymer composed of styrene monomers. Polymers obtained synthetically, based on oil or petroleum, are commonly referred to as plastics or rubbers. In contrast, there is also a wide variety of polymers found in nature, usually denominated by biopolymers, such as cellulose, chitosan, genetic material, and carbohydrates.

Polymers consisting of only one type of monomer are classified as homopolymers. Otherwise, they are classified as copolymers, which in turn can be grouped according to the arrangement of the monomers in the polymer chain [1] (random, alternating, block, and graft are the usual arrangements). Polymers are also classified as unbranched, when consisting of a simple linear chain, or branched, when comprising a primary chain with one or more secondary side chains [1]. Moreover, polymers can also exhibit complex and intricate microscopic structures due to the arrangements of the chains, which can be chemically bonded, forming cross-linked structures. In contrast, semi-crystalline structures form

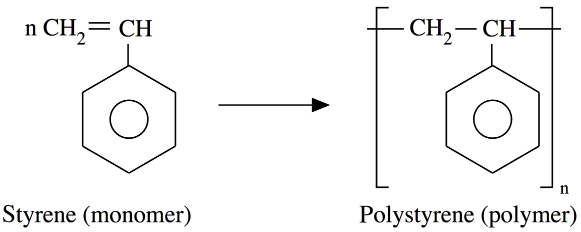
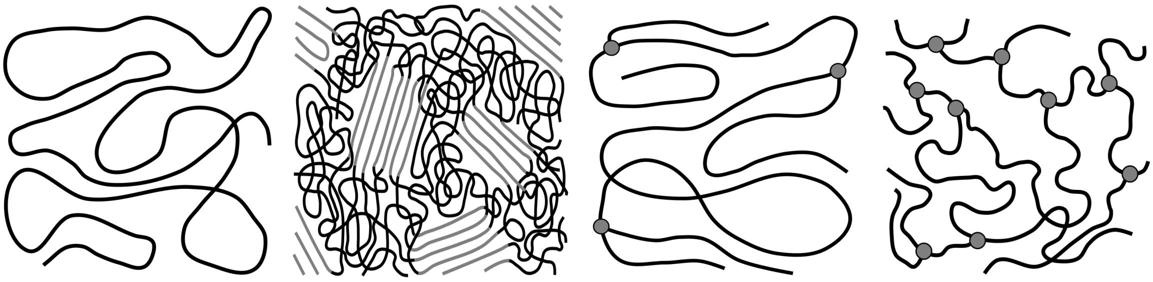


Figure 1.1: Polystyrene composed of styrene monomers (adapted from R.O. Ebewele, Polymer science and technology, CRC press, 2000).

when individual chains are folded and packed regularly in three-dimensional arrangements, known as crystallites [1]. The microscopic arrangement of the chains, which are illustrated in Figure 1.2, strongly determines the materials physical properties. In the solid-state, slightly cross-linked polymers tend to form elastomers, also referred to as rubbers, whereas highly cross-linked polymers tend to form thermoset plastics, which are irreversibly hardened. On the other side, semi-crystalline polymers tend to form opaque materials, whereas amorphous polymers tend to be transparent. In both cases, these polymers can be melted and solidified repeatedly, which justifies their group denomination as thermoplastics. Molten thermoplastics also exhibit unique and, sometimes, counter-intuitive rheological properties, such as a combination of viscous and elastic behaviours, referred to as viscoelasticity. Indeed, the viscosity is usually associated with fluids, whereas the elasticity is typically associated with solids. The complexity of polymers has enabled the development of materials with advantageous physical properties, such as lightness and robustness, relevant for many technological fields, such as the transportation industry, health, construction, packaging, among many others.



(a) Amorphous. (b) Semi-crystaline. (c) Slightly cross-linked. (d) Highly cross-linked.

Figure 1.2: Typical microscopic structures of polymers.

1.1.2 Extrusion process

Several manufacturing technologies are used in the polymer processing industry to process thermoplastics, which mainly depend on the characteristics of the desired product, particularly on its geometry. Standard manufacturing technologies include the extrusion (pushing the molten polymer through a die), the injection moulding (injecting the molten polymer into a cavity), the extrusion blow moulding (inflating a molten polymer tube inside a mould to create a hollow part), the 3D printing (depositing a molten polymer layer by layer to form the part), and the thermoforming (using suction to force a heated polymer sheet into a mould). A comprehensive literature on these manufacturing technologies and their application is found in M.G. McGrum et al., 1997 [2], D.M. Bryce, 1997–1999 [3–6], T.A. Osswald et al., 2006 [7], Z. Tadmor et al., 2006 [8], S. Thomas et al., 2009 [9], O.S. Carneiro et al., 2012 [10], and D.G. Baird et al., 2014 [11]. In the present work, a particular focus is provided for the extrusion process, which has a relevant role in nowadays polymer processing industry.

Constant cross-section items, such as tubing, sheets, films, and also structural components, such as profiles for window frames, are usually manufactured by extrusion. An illustration of a typical thermoplastic profile extrusion line is provided in Figure 1.3. The process is continuous and consists of an extruder, which comprises a screw rotating inside a heated barrel, and is gravity-fed with thermoplastic pellets from a top-mounted hopper, placed at the rear. The rotating screw compresses and forces the raw polymeric material to move forward through the barrel, in which several independently controlled heating units, mounted in sequence, gradually increase the temperature necessary for the melting. The viscous dissipation is another important thermodynamic phenomenon, responsible for generating the heat required to melt the polymer. However, it makes more difficult the temperature control and increases the risk of overheating. The extrusion die is mounted at the front of the barrel, and its cross-section corresponds to the desired product cross-section, through which the molten polymeric material, subject to compression, is forced to flow.

After leaving the extrusion die, the still molten product is cooled and geometrically calibrated to guarantee the desired shape on the solidified product. Pipes are usually cooled with chilled water baths inside sealed chambers subject to a carefully controlled vacuum that avoids the molten polymer from collapsing or deforming, before solidification. Other profiles, such as structural components, are usually cooled in contact with metallic systems, having an internal chilled water system, which also calibrates the product relevant dimensions, as illustrated in Figure 1.4. Subsequently, a caterpillar haul-off, composed of puller rolls, imposes the desired linear production velocity to the extruded product. A typical extrusion line ends with a cutting unit, which cuts the profile into predefined lengths.

1.1.3 Industrial challenges

Polymer processing industries are highly sophisticated with advanced, expensive, and highly-automated manufacturing technologies. However, the complex behaviour of polymers and the complex thermo-mechanical processes involved in these manufacturing technologies, raise challenging engineering problems, even for experienced polymer engineers. Some of these challenges in the case of the extrusion are described hereafter.

The flow balance of the molten polymeric material at the die openings is crucial to avoid that more material flows through the thicker sections of the profile, when compared with the thinner ones [12]. Therefore, an unbalanced flow does not allow to obtain the shape desired for the final

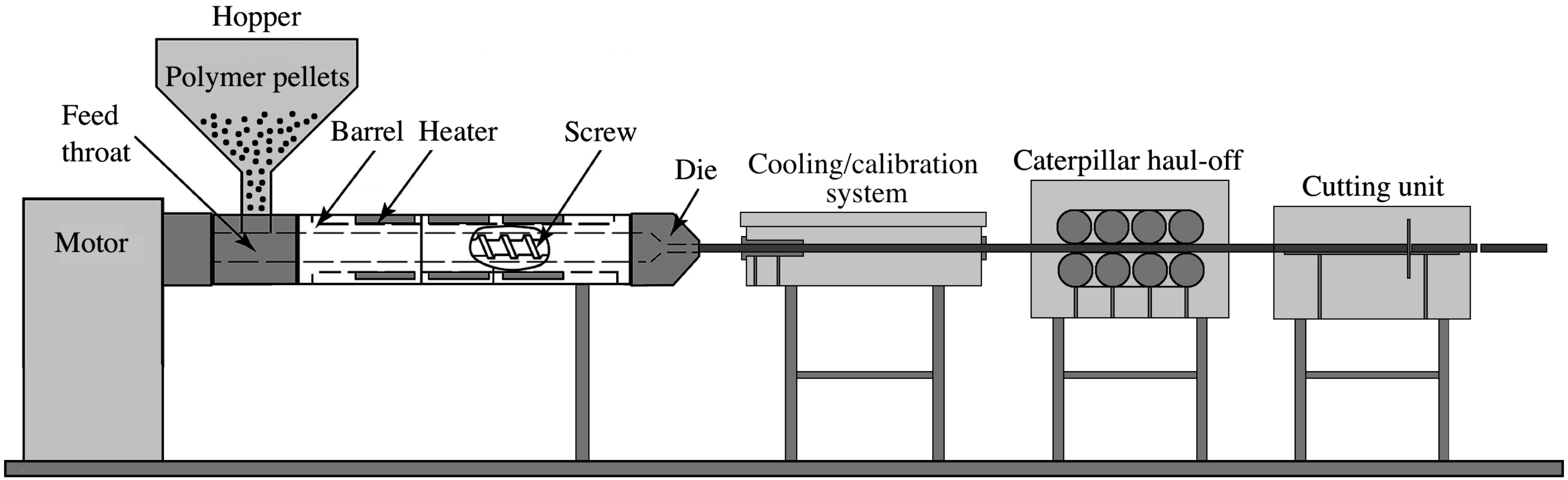


Figure 1.3: Typical thermoplastic profile extrusion line (adapted from C. Rauwendaal, R. Gonzalez-Nunez, D. Rodrigue, Polymer processing: extrusion, in Encyclopedia of polymer science and technology, John Wiley & Sons, 2017).

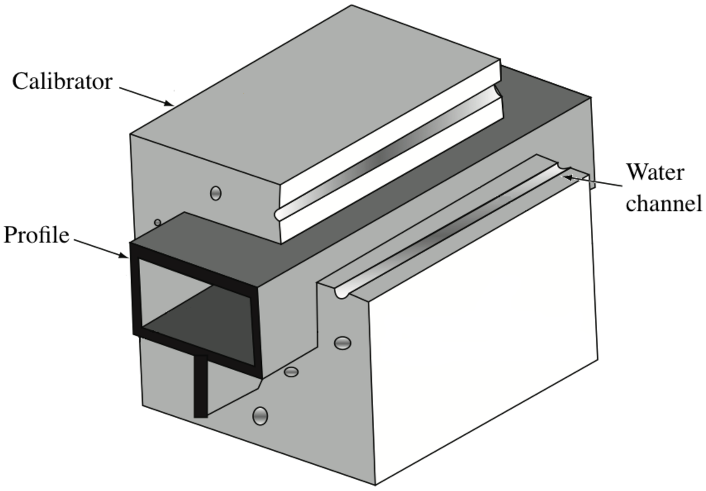
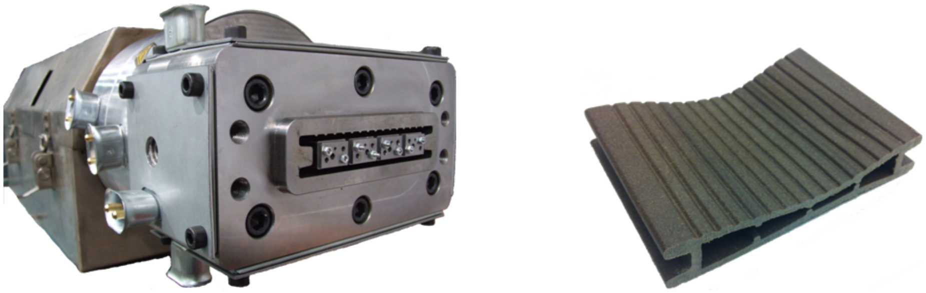


Figure 1.4: Example of a cooling/calibrator system for the thermoplastic profile extrusion (adapted from O.S. Carneiro, J.M. Nóbrega, Design of extrusion forming tools, Smithers Rapra, Shawbury, 2012).

solidified product, as exemplified in Figure 1.5 for the case of a deck. The usual practice to ensure a balanced flow consists in gradually adjusting the die geometry from the barrel to the openings, such that the thinner sections of the profile are compensated with more material. However, due to the complex behaviour of the molten polymeric material, the effect of these adjustments in the flow is difficult to predict through pure analytic approaches mainly due to the large number of variables involved in the process. On the other side, the common trial-and-error approach requires large amounts of resources, is very time-consuming, and might limit the possibility to achieve optimal configurations. Indeed, the die design is a challenging task, even for experienced polymer engineers, leading to increased costs.


(a) Extrusion die.

(b) Defective profile.

Figure 1.5: Extrusion die for the production of deck and defective profile resulted from an unbalanced flow (adapted from L. Ferrás, Theoretical and numerical studies of slip flows, PhD Thesis, Universidade do Minho, 2012).

Another challenge of great concern in the extrusion, which also applies other different manufacturing technologies, is the cooling of the molten polymeric material after leaving the die [12]. The cooling rate strongly determines the structure of the polymeric material and, thus, affects the physical properties of the final solidified product, such as density, optical and barrier properties, coefficient of friction, and impact behaviour. In that regard, a high cooling rate is usually desired, particularly in the case of sheets and films, avoiding large crystallites and, consequently, providing smoothness and transparency. Moreover, fast cooling is also desired for increasing the production and profitability of the process. On the other side, the average temperature on the profile cross-section must fall below the solidification point of the polymeric material after the cooling stage. Unfortunately, a high cooling rate might result in solidified borders and molten cores, especially in thicker sections, which might induce subsequent remelting of the product due to heat conduction. Moreover, different cooling rates result in internal residual stresses, which also lead to subsequent deformation of the final solidified product. In that regard, a uniform cooling rate is desired across the profile cross-section, which in general conflicts with a high cooling rate.

The previous challenge leads to an optimization problem, for which several parameters have to be considered, as illustrated in Figure 1.6, namely, the extrusion conditions, cooling conditions, profile geometry, system geometry, polymer properties, metal properties (in the case of structural components), and the vacuum conditions (in the case of tubing and pipes), among others. In the case of structural components, the optimization often consists in performing adjustments to the cooling/calibrator system geometry, configuration, or cooling conditions. As in the previous case, the optimization problem is commonly performed with trial-and-error approaches implying the same drawbacks, which often leads to intractable challenges. Similar challenges arise in the case of the injection moulding, where most of the residence time is dedicated to the cooling stage and, therefore, optimizing the heat transfer under the same principles is of crucial importance.

In light of the drawbacks of trial-and-error approaches in the context of these industry challenges, there is a deep concern for developing innovative and efficient alternative design approaches. In that regard, the emerging field of computational modelling has received increasing attention from the industry side to provide means of overcoming these challenges and effectively support design activities.

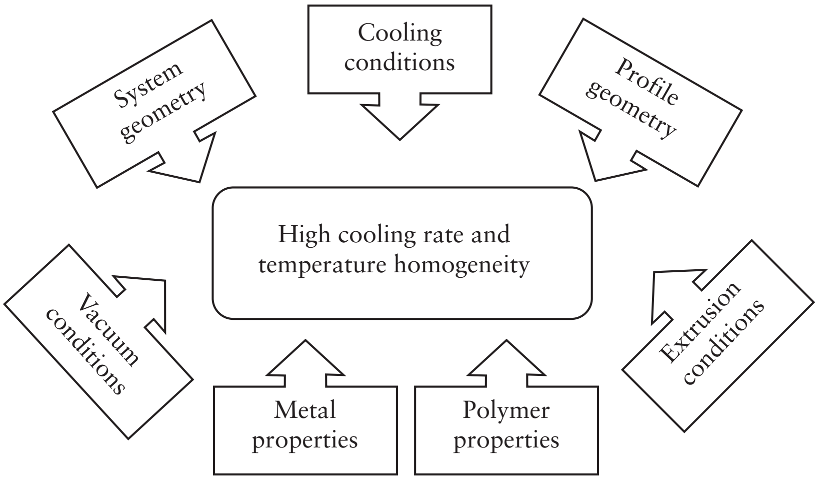


Figure 1.6: Parameters affecting the optimization of the extrusion cooling stage (adapted from O.S. Carneiro, J.M. Nóbrega, Design of extrusion forming tools, Smithers Rapra, Shawbury, 2012).

1.2 Computational modelling

The computational modelling is a multidisciplinary field involving physics, mathematics, and computer science, which has enabled overcoming ever-increasing complexity of engineering problems in the last decades. The general methodology consists in using mathematical models to describe the physical phenomena in engineering problems, which are then solved with the use of proficient numerical methods and advanced computer algorithms. In that regard, the computational modelling has become a comprehensive and self-contained scientific discipline, which nowadays is subject of extensive research in many fields of natural sciences and engineering. The computational modelling has known an expanding application in many research fields, as happens in the polymer processing industry, which has been also taking advantage of this powerful tool to overcome the drawbacks of the usual trial-and-error approaches [13–22]. The present section describes the general methodology of computational modelling, whereas the subsequent sections provide relevant information concerning the mathematical models and the numerical methods.

1.2.1 General methodology

The computational modelling has seen constant development from its inception. Since then, innovative approaches have continuously emerged to address the needs and limitations of the industry in solving their problems. The general methodology of the computational modelling is comprehensively described in the book of M. Schäfer, 2006 [23]. Additionally, the books of R.R. Huilgol et al., 1997 [24], R.G. Owens et al., 2002 [25], and M.J. Crochet et al., 2012 [26], for instance, provide a particular focus on the computational modelling applied to fluid mechanics and rheology, as applied in polymer processing applications. A schematic view of the general methodology is illustrated in Figure 1.7, which is briefly described hereafter.

The first step consists in defining the geometry and the mathematical models for the physical phenomena involved in the problem of interest. For instance, constitutive equations derived from rheological principles describe the phenomenological relationships between mechanical variables (and eventually also thermal variables) of fluids in motion. On the other side, governing equations derived from the solid or fluid mechanics theories model the heat and mass transfer according to the materials physical properties. Boundary conditions are also required to impose the physically sound limits of the problem unknowns, whereas interface conditions impose the interaction between different materials or regions in contact. These mathematical models comprise several unknown physical variables, necessary to quantitatively describe the physical phenomena, such as temperature, velocity, stress, and pressure. Moreover, they were were initially derived from an analytic and experimental process, where conservation Laws or principles were gradually developed and adapted in an attempt to explain the experimental observations. However, the comprehensive literature available nowadays on mathematical models, also under continuous evolution, allows engineers to adequately describe the

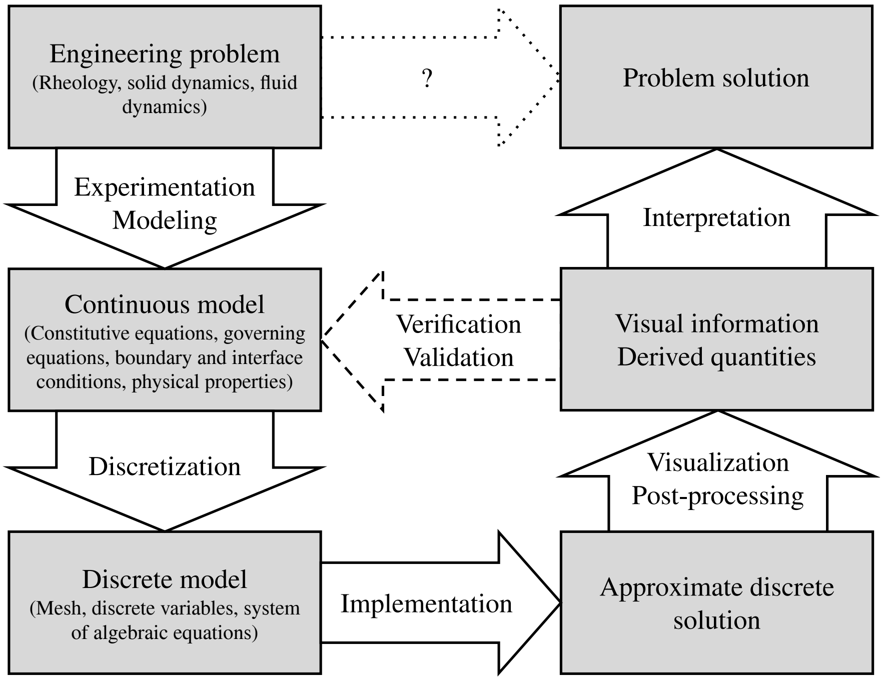

Engineering problem
(Rheology, solid dynamics, fluid dynamics)
?
Problem solution
Experimentation
Modeling
Interpretation
Continuous model
(Constitutive equations, governing equations, boundary and interface conditions, physical properties)
Verification
Validation
Visual information
Derived quantities
Discretization
Visualization
Post-processing
Discrete model
(Mesh, discrete variables, system of algebraic equations)
Implementation
Approximate discrete solution

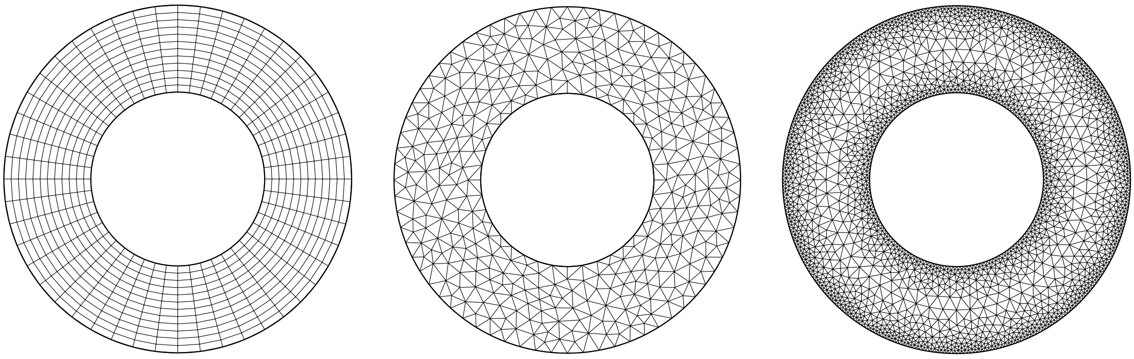
Figure 1.7: The general methodology of the computational modelling for the solution of engineering problems (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

physical phenomena in a wide range of problems. Similarly, the physical properties of the materials are usually determined based on experimentation, for which extensive documentation is also available for most materials [27, 28].

The mathematical models are often complex, comprising systems of partial differential equations intractable analytically when considering geometries with practical interest (otherwise, the computational modelling methodology would be unnecessary). In that regard, proficient numerical methods were developed to solve these mathematical models and provide an approximate solution to the associated unknown physical variables. The usual strategy consists in transforming the continuous model into a discrete model, referred to as discretization, which can be solved using algebra techniques and advanced computer algorithms. This discretization procedure is performed at two levels, firstly at the level of the problem geometry, and secondly at the level of the mathematical models, as detailed hereafter.

In the first part of the discretization, mesh generation techniques subdivide the geometry into a contiguous mesh of discrete elements, usually consisting of triangles or quadrilaterals (in the two-dimensional case), although any element type can be considered. Meshes are often classified as structured or unstructured according to the spatial arrangement of the cells, as illustrated in Figure 1.8. In structured meshes, the arrangement of the cells is regular, which often leads to more efficient computer memory management and usage, due to the trivial connectivity between neighbour mesh elements. Moreover, mesh generation algorithms for structured meshes are simple to implement and computationally efficient, although they become cumbersome, or impossible, to adapt and apply in complex geometries, as those arising in typical engineering problems. In contrast, unstructured meshes have an irregular arrangement of the cells, which makes the mesh generation simpler to adapt to complex geometries. However, more elaborated algorithms must be implemented, often leading to more computationally expensive memory management and usage, and there is a wide range of open-source software available for mesh generation purposes. Another advantage of unstructured meshes is the greater ease of creating local refinements, often demanded to increase the accuracy of the solution in critical (large gradient) regions.

The second part of the discretization concerns the partial differential equations in the mathematical models and the associated physical variables defined for the whole problem geometry. One common approach is to represent the physical variables in the continuous problem as piece-wise numerical variables, usually associated to the cells or the vertices, or even both, depending on the technique. These variables correspond to unknowns of the discrete model, which is derived from a discretization method used to provide algebraic approximations to the partial differential equations in each reference mesh element, usually the cells or the vertices. In that regard, a linearization process is usually necessary in the case of non-linear models before the discretization method is applied. The algebraic equations composing the discrete model can be understood as local representations of the physical phenomena, being more meaningful or more abstract according to the discretization method. In any case, each equation relates linearly a set of discrete variables in the vicinity of the reference mesh element, which ultimately connects all the equations due to common discrete variables. There are many discretization methods, where the choice depends on many factors, such as the partial differential equations, the type and arrangement of the mesh elements, or the numerical properties of the method. For instance, some discretization methods can take advantage of orthogonal structured meshes, that is, when the cell vertices are aligned in orthogonal coordinates, leading to more straightforward derivations and more computationally efficient simulations. Finally, regardless of the discretization method, the discrete model usually consists of a large number of algebraic equations for the discrete variables, which are then assembled in a system of linear equations.



(a) Structured quadrilateral mesh.

(b) Unstructured triangular mesh.

(c) Refined unstructured mesh.

Figure 1.8: Types of meshes for an annular domain.

The system of linear equations is solved with matrix algebra techniques, which provides an approximate solution of the physical variables in the continuous problem in terms of numerical variables. In practical problems, meshes consisting of thousands or even hundreds of millions of elements are used to provide enough accuracy and, therefore, a large number of numerical variables are computed in the discrete model. In that regard, the computational aspects become much more relevant, and the implementation with advanced computer algorithms is crucial in the quest for efficient simulations, whose results are obtained in a reasonable time frame. The approximate solution is not intuitively understood by looking at such a large number of variables, consequently suitable visualization software with post-processing capabilities is used to provide visual information and derive other physical quantities. Finally, the data is interpreted in the context of the problem, providing means of evaluating and addressing the shortcomings of the application, for instance, to improve the process efficiency.

1.2.2 Validation and verification

The application of the computational modelling allows obtaining an approximate solution for the problem, which is, therefore, prone to errors that should be investigated in the quest for a reliable methodology. In that regard, the validation of the methodology inspects the quality of the mathematical models in describing the physical phenomena, whereas the verification concerns the quality of numerical methods in solving the mathematical models. Besides the modelling error, two primary numerical sources are contributing to the error of the approximate solution, as illustrated in Figure 1.9. The first error source is the discretization of the continuous model, which results in a system of linear equations that approximates the partial differential equations locally in the mesh elements. Therefore, assuming that the solution of the continuous model can be determined analytically, and the solution of the system of linear equations is computed exactly, there would be still a discretization error. In that case, the accuracy strongly depends on the mesh and the discretization method. On the other side, the solution of the system of linear equations cannot be computed exactly due to the truncated representation of real numbers in computers, which is also a source of rounding errors. Moreover, iterative matrix algebra techniques are often used to accelerate the computation process, for which the solution is usually computed, satisfying a residual tolerance. Indeed, the total numerical error is a combination of the discretization error and the computation error.

The common practice of verification consists in using specific techniques to assess the behaviour of the numerical method, namely in terms of consistency, accuracy, convergence, robustness, and stability. Analytical techniques are ideally employed to derive or estimate these properties, whereas such approach becomes cumbersome, or even infeasible, for some numerical methods due to its characteristics. In that regard, numerical benchmarks are usually employed, where several test cases are manufactured with specific difficulties to check the capabilities of the numerical method. Consistency, accuracy, convergence, robustness, and stability are then evaluated through some kind of error inspection of the approximate solution under mesh refinement. After checking that the numerical method is capable of solving the model appropriately, the validation phase usually consists in comparing the approximate solution with experimental measurements. In that regard, the mathematical models are questioned and revised accordingly, often choosing different constitutive or governing equations or even determining the response of the physical properties under different situations.

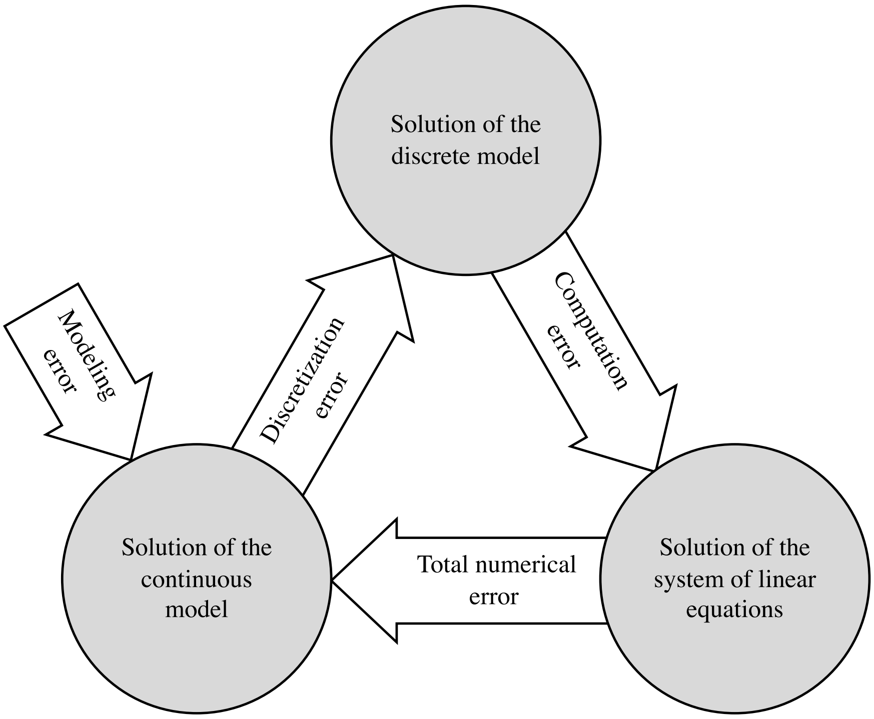

Solution of the discrete model
Modeling error
Discretization error
Computation error
Solution of the continuous model
Total numerical error
Solution of the system of linear equations

Figure 1.9: The general methodology of the computational modelling for the solution of engineering problems (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

1.2.3 Advantages and limitations

The computational modelling has become a fundamental instrument in many research fields, which includes the polymer processing industry. Indeed, the benefits of using the computational modelling approach to complement the common trial-and-error experimental based approaches are remarkable. Firstly, several scenarios of system geometries, processing conditions, cooling conditions, physical properties, among others, can be virtually investigated in the quest of an optimal processing configuration and operation. Consequently, it reduces the amount of material, time, and money usually required in trial-and-error experimental based approaches, whereas the use of nowadays powerful computers allows finding more efficiently the optimal processing configuration. Another advantage of the computational modelling approach is the possibility to assess virtually the physical variables at any location, including those that are usually inaccessible through experimentation. Moreover, it often gives a more comprehensive insight into the physical phenomena due to the simultaneous assessment of several physical variables and derived quantities relevant to the problem. Despite these benefits, experimental measurements are always required to validate both the mathematical models and the computed results. In that regard, the development of experimental techniques is an ideal and necessary complement for the practical applicability and utility of the computational modelling approach in the industry.

1.3 Mathematical modelling

The physical phenomena involved in the polymer processing applications of interest are mathematically modelled in the present section such that the computational modelling approach can be applied, for which the extrusion process is taken as an example. In that regard, the mathematical models derive from several theories, namely the rheology to provide the constitutive equations for the constitutive modelling of polymeric fluids, and the fluid mechanics to provide the governing equations for the heat and mass transfer modelling. A comprehensive description of the mathematical modelling of polymeric fluid flows is found, for instance, in the books of R.B. Bird et al., 1987 [29], C.W. Macosko, 1994 [30], R.G. Larson, 1999 [31], and F.A. Morrison, 2011 [27], which is briefly described as hereafter.

1.3.1 Constitutive modelling

The rheology is a well-established science for a wide range of materials, particularly polymeric materials, dedicated to the study of deformation and fluid flow involving a viscous or viscoelastic response to stress. Constitutive equations for polymeric fluids are often based in rheological principles to describe the phenomenological relationships between mechanical variables (and eventually also thermal variables) mathematically. Newton's law of viscosity states that the stress of a fluid in motion changes linearly with the strain rate, where the proportionality constant is referred to as dynamic or absolute viscosity. Many examples of these Newtonian fluids are present everywhere, such as water, mineral oil, and gas. However, the complex nature of polymers makes the stress in polymeric fluid flows exhibit a non-trivial response to the strain rate, usually significantly deviating from Newton's law of viscosity. In that regard, more complex constitutive equations for these non-Newtonian fluids are needed, often derived from empirical relations within experimentation rather than based on fundamental physical principles.

Several theories for the rheology of non-Newtonian fluids attempt to adequately describe the non-trivial behaviour of polymeric fluids, providing constitutive models that gradually increase in complexity. The most straightforward approach is the generalized Newtonian fluid, which consists in replacing the absolute viscosity with an apparent or effective viscosity, which is a function of the second invariant of the strain rate tensor. Empirical relations, such as the classical power-law, are then used to quantitatively describe the non-linear response of the stress to the strain rate, for which experiments are required to determine the characteristic parameters in the empirical constitutive relation for the fluid. The generalized Newtonian fluid approach becomes convenient and straightforward, providing valid results in a variety of applications, particularly those comprising slow processes, when comparing the process characteristic time with the material relaxation time. Many non-Newtonian fluids exhibiting different behaviours are appropriately modelled as generalized Newtonian fluids. For instance, shear-thinning or shear-thickening fluids, where increasing stress leads to the decrease and increase of the apparent viscosity, respectively. Another example is thixotropic and rheopecty fluids, showing a time-dependent viscous change, where the duration of the applied stress decreases and increases the apparent viscosity, respectively. However, since the strain rate history is not considered, the generalized Newtonian fluid approach only reproduces a viscous response, also referred to as inelastic, whereas polymeric fluids exhibit a combination of viscous and elastic responses. Consequently, the generalized Newtonian fluid approach might inadequately describe the behaviour of these fluids in polymer processing applications.

The constitutive modelling for viscoelastic responses consists in splitting the stress tensor into a solvent contribution and a polymer solute contribution, also referred to as elastic or polymeric stress. The solvent is assumed to have a Newtonian behaviour, whereas the polymer solute requires complex partial differential equations that comprise several relaxation modes, due to the length distribution of the polymer chains. From linear to non-linear viscoelastic fluids, several constitutive models attempt to reproduce better the observed behaviour of complex polymeric fluids, which unfortunately comes at the cost of increasing complexity. Examples of empirical relations for the constitutive modelling of fluids are provided in Figure 1.10. For the sake of completeness, inviscid fluids are mentioned, corresponding to those having a null viscosity and, consequently, no stress, and are of interest in other contexts, such as aerospace applications.

1.3.2 Heat and mass transfer modelling

Besides the constitutive modelling of the polymeric fluids in polymer processing applications, the extrusion process consists of two fundamental physical phenomena, namely the heat and mass transfer. Governing equations for the heat and mass transfer are required, usually involving several physical variables, namely the temperature, velocity, pressure, and stress, whereas the density is also considered in compressible fluid flows. In that context, the computational fluid dynamics has emerged as a branch of the computational modelling dedicated to the heat and mass transfer modelling in fluid flow problems. For fluids with non-Newtonian behaviour, such as polymeric fluids, which require complex constitutive equations derived from the rheology theory, this field is often referred to as the computational rheology.

The compressibility effects during the extrusion process, for instance, considering the molten polymeric material flowing through the die, can be neglected and, therefore, the density is a physical property rather than a physical variable. Moreover, the extrusion process is continuous, and the physical variables do not usually change over time and, therefore, a steady-state situation can be

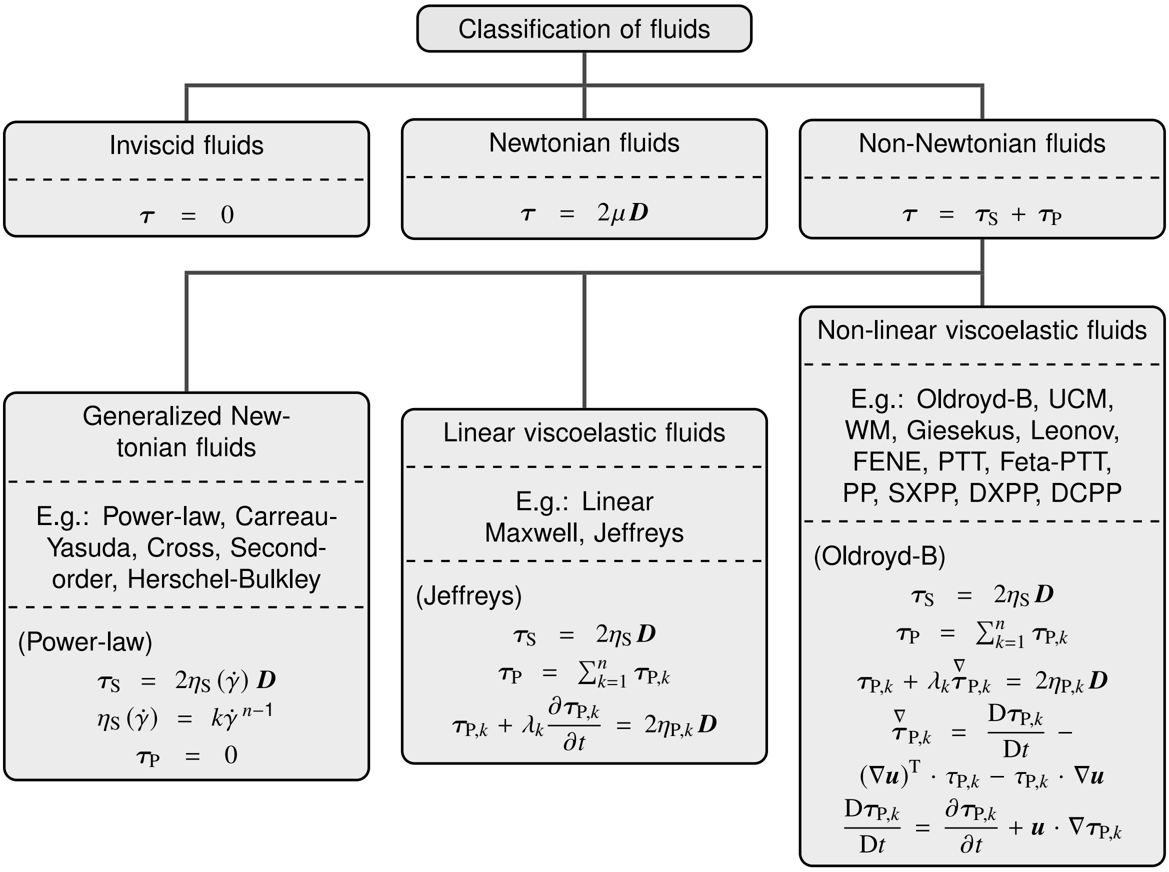

Classification of fluids
Inviscid fluids
$\boldsymbol{\tau} = 0$
Newtonian fluids
$\boldsymbol{\tau} = 2\mu \boldsymbol{D}$
Non-Newtonian fluids
$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{S}} + \boldsymbol{\tau}_{\mathrm{P}}$
Generalized Newtonian fluids
E.g.: Power-law, Carreau-Yasuda, Cross, Second-order, Herschel-Bulkley
(Power-law)
$\boldsymbol{\tau}_{\mathrm{S}} = 2\eta_{\mathrm{S}}(\dot{\gamma})\boldsymbol{D}$
$\eta_{\mathrm{S}}(\dot{\gamma}) = k\dot{\gamma}^{n-1}$
$\boldsymbol{\tau}_{\mathrm{P}} = 0$
Linear viscoelastic fluids
E.g.: Linear Maxwell, Jeffreys
(Jeffreys)
$\boldsymbol{\tau}_{\mathrm{S}} = 2\eta_{\mathrm{S}}\boldsymbol{D}$
$\boldsymbol{\tau}_{\mathrm{P}} = \sum_{k=1}^{n} \boldsymbol{\tau}_{\mathrm{P},k}$
$\boldsymbol{\tau}_{\mathrm{P},k} + \lambda_k \frac{\partial \boldsymbol{\tau}_{\mathrm{P},k}}{\partial t} = 2\eta_{\mathrm{P},k}\boldsymbol{D}$
Non-linear viscoelastic fluids
E.g.: Oldroyd-B, UCM, WM, Giesekus, Leonov, FENE, PTT, Feta-PTT, PP, SXPP, DXPP, DCPP
(Oldroyd-B)
$\boldsymbol{\tau}_{\mathrm{S}} = 2\eta_{\mathrm{S}}\boldsymbol{D}$
$\boldsymbol{\tau}_{\mathrm{P}} = \sum_{k=1}^{n} \boldsymbol{\tau}_{\mathrm{P},k}$
$\boldsymbol{\tau}_{\mathrm{P},k} + \lambda_k \overset{\nabla}{\boldsymbol{\tau}}_{\mathrm{P},k} = 2\eta_{\mathrm{P},k}\boldsymbol{D}$
$\overset{\nabla}{\boldsymbol{\tau}}_{\mathrm{P},k} = \frac{\mathrm{D}\boldsymbol{\tau}_{\mathrm{P},k}}{\mathrm{D}t} - (\nabla\boldsymbol{u})^{\mathrm{T}} \cdot \boldsymbol{\tau}_{\mathrm{P},k} - \boldsymbol{\tau}_{\mathrm{P},k} \cdot \nabla\boldsymbol{u}$
$\frac{\mathrm{D}\boldsymbol{\tau}_{\mathrm{P},k}}{\mathrm{D}t} = \frac{\partial \boldsymbol{\tau}_{\mathrm{P},k}}{\partial t} + \boldsymbol{u} \cdot \nabla\boldsymbol{\tau}_{\mathrm{P},k}$

Figure 1.10: Classification and constitutive modelling of fluids.

considered. In that regard, the governing equations for the heat and mass transfer consist of a heat equation, a momentum equation, and a continuity equation, which are illustrated in Figure 1.11 (T is the temperature, $\boldsymbol{u}$ is the velocity, p is the pressure, $\boldsymbol{\tau}$ is the stress, $\boldsymbol{I}$ is the identity matrix, κ is the thermal conductivity, ρ is the density, c_{P} is the specific heat capacity, and μ is the absolute viscosity).

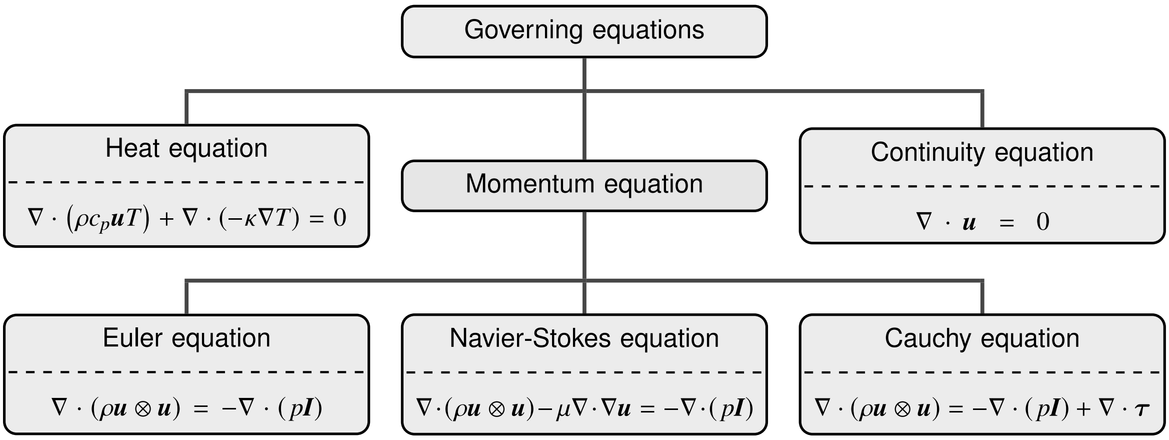

Governing equations
Heat equation
$\nabla \cdot (\rho c_p \boldsymbol{u} T) + \nabla \cdot (-\kappa \nabla T) = 0$
Momentum equation
Continuity equation
$\nabla \cdot \boldsymbol{u} = 0$
Euler equation
$\nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla \cdot (p\boldsymbol{I})$
Navier-Stokes equation
$\nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) - \mu \nabla \cdot \nabla \boldsymbol{u} = -\nabla \cdot (p\boldsymbol{I})$
Cauchy equation
$\nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla \cdot (p\boldsymbol{I}) + \nabla \cdot \boldsymbol{\tau}$

Figure 1.11: Governing equations for steady-state incompressible flows.

The mass transfer applies only to fluids and requires two governing equations, one for the conservation of mass and another for the balance of momentum of the fluid, also referred to as continuity and momentum equations, respectively. The continuity equation imposes a null divergence velocity, which guarantees conservation of mass in the system for incompressible fluid flows, whereas the choice of the momentum equation mainly depends on the fluid. The Euler equation involves only the velocity and the pressure to describes the balance of momentum in inviscid fluid flows and, therefore, is not appropriate to polymer processing applications. On the other side, the Navier-Stokes equation is usually used to describe the balance of momentum in Newtonian fluid flows, providing the absolute viscosity of the fluid. For non-Newtonian fluid flows, the Cauchy equation has to be used instead, which becomes more complicated since the contribution of the stress is required in the momentum equation, obtained from the corresponding constitutive equations for the fluid. However, when the constitutive modelling is provided with explicit empirical relations, as for generalized Newtonian fluids, the contribution of the stress can be rewritten in terms of the effective viscosity, which becomes similar to the Navier-Stokes equation.

From the mass transfer modelling viewpoint, explicit relations between stress and strain rate lead to a model consisting of the Navier-Stokes equation and the continuity equation. On the other side, implicit relations require the use of the Cauchy equation and, consequently, the stress has to be determined from the constitutive equations that are part of the model. Moreover, the constitutive modelling often consists of several equations, which in turn unfold into more equations since they are usually written in the tensor form but are solved for each tensor component. For instance, assuming a tridimensional problem, the momentum equation unfolds into three equations derived for each velocity component, whereas the continuity equation is already in the scalar form. On the other hand, each constitutive equation unfolds into six equations, assuming a symmetric stress tensor, or nine equations otherwise. Therefore, the use of implicit constitutive models increases considerably the complexity of the mass transfer modelling, which ultimately raises essential challenges from the numerical and computational viewpoints.

The heat transfer equation describes the energy transport in a solid or fluid material, which occurs through convection and conduction, whereas radiation is usually neglected in polymer processing applications, such as extrusion and injection moulding. The heat transfer by convection is associated with the motion of mass and, therefore, applies to molten polymeric materials and also to moving parts of the extrusion machine. The heat convection is proportional to the temperature and the velocity, which are both physical variables of the problem, and also to the density and the specific heat capacity, which are physical properties of the material. On the other side, the heat transfer by conduction is associated with the random propagation of the microscopic motion of molecules or atoms in favour of decreasing temperature gradients. In that case, the thermal conductivity of the material measures the associated rate of heat transfer. Thus, the heat conduction applies not only to molten polymeric materials and moving parts of the extrusion machine but also to all components, such as the cooling/calibration system.

The heat transfer equation can only be solved after the solution of the balance of momentum and continuity equations since the convective transport of energy depends on the velocity. However, in practice, the opposite is also valid as the physical variables for the mass transfer, namely the density and the viscosity or associated constitutive equations for the material, often depend on the temperature. Indeed, changes in the physical state also occur during the extrusion process, from the raw polymeric material melting inside the barrel to the final product solidification at the cooling/calibration system. In that regard, large temperature variations throughout the process are required, making the dependency of these physical properties on the temperature more relevant. Additionally, the thermal viscous dissipation is another relevant phenomena in polymer processing applications resulting from the typically high stress, which transforms mechanical energy into thermal energy. In practical terms, the thermal aspects in the extrusion process are relevant for an appropriate mathematical model. Therefore, the mathematical modelling for polymer processing applications results in non-linear systems of partial differential equations, which cannot be solved for each equation separately.

Boundary conditions complement the governing equations, constraining the physical phenomena on the boundaries of the problem, such as the temperature, the conductive heat flux, the velocity, among others. Indeed, several types of boundary conditions can be prescribed for the heat and mass transfer, where few examples are illustrated in Figure 1.12, for the mathematical modelling of a cooling/calibration system in the extrusion. Moreover, interface conditions are required to impose the interaction between different regions in contact, for instance, between the molten polymeric material and the cooling/calibrator system. In the case of the heat transfer, both the continuity of the temperature and the conservation of the conductive heat flux are usually prescribed, providing valid results in a variety of situations. However, this idealization of a perfect thermal is not appropriate in many applications since realistic contacts are often composed of microscopic air pockets due to the surface roughness, particularly in solids, as illustrated in Figure 1.13. In these imperfect thermal contacts, a relevant interfacial thermal resistance arises, which hampers the heat exchange between the polymer melt and the metallic parts. In that case, temperature jumps are experimentally measured on the interface, as in the case between the molten polymeric material and the cooling/calibration system in the extrusion. Therefore, specific interface conditions are required to replicate this behaviour appropriately. This whole range of issues in the extrusion process highlights the importance of properly accounting for the thermal effects in the computational modelling of polymer processing applications.

1.4 Computational methods

In the multidisciplinary field of the computational modelling, the development of accurate and efficient numerical methods has been a topic of extensive research. In that regard, uncountable spatial discretization methods have emerged, where three commonly used classes of methods are relevant to mention, namely the finite difference method, the finite element method, and the finite volume method. Notice that time discretization methods for unsteady mathematical models are not in the scope of the present work. In any case, as described in Section 1.2, the general strategy consists

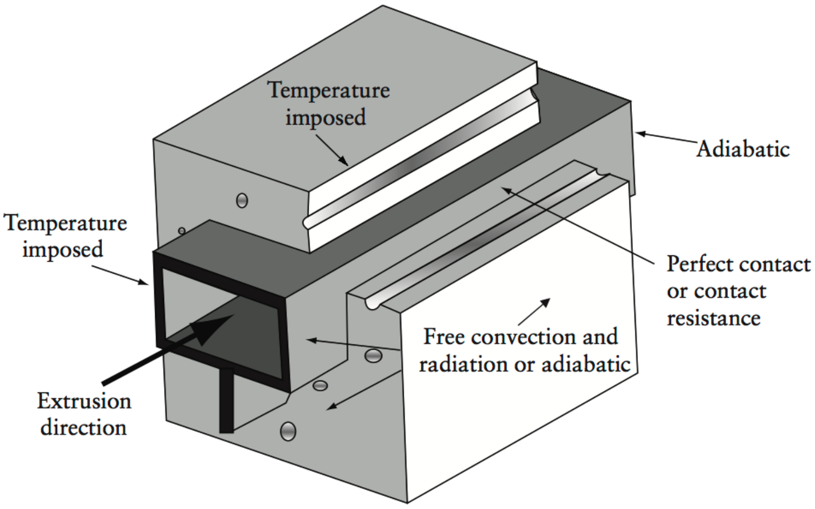

Temperature imposed
Adiabatic
Temperature imposed
Perfect contact or contact resistance
Free convection and radiation or adiabatic
Extrusion direction

Figure 1.12: Boundary and interface conditions prescribed for the thermoplastic profile extrusion cooling stage with a cooling/calibrator system (adapted from O.S. Carneiro, J.M. Nóbrega, Design of extrusion forming tools, Smithers Rapra, Shawbury, 2012).

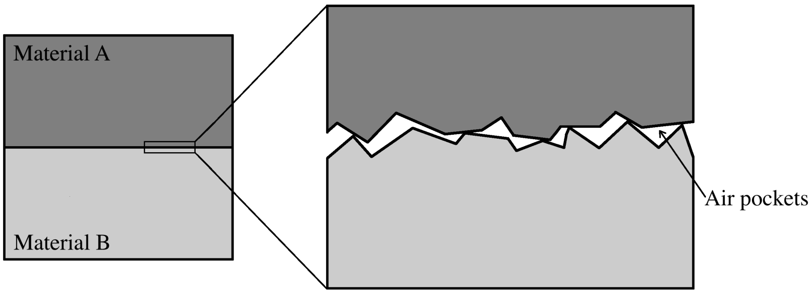


Figure 1.13: Enlargement of the contact at the microscale between two materials.

in transforming the partial differential equations into a system of algebraic equations, each relating a set of local mesh variables, which is then solved with matrix algebra techniques. These methods are briefly described hereafter, where a particular focus is given to the finite volume method, which is the approach employed in the present work.

1.4.1 Discretization methods

The finite difference method is one of the simplest and oldest discretization technique, which has been historically used in the context of computational fluid dynamics problems [32–34]. The discretization technique in the finite difference method is based on a truncated Taylor series expansion to approximate the derivatives in the partial differential equations in each cell of the mesh. A derivative of a specific order, in a truncated Taylor series, is approximated as a difference between the derivatives of previous order at the cell points, divided by the distance between the cell points. Similarly, the derivatives in the partial differential equations are approximated in terms of discrete variables in the vicinity of each cell. Then, all the equations are assembled in a system of linear equations, which is solved with matrix algebra techniques. Although the derivation of the finite difference method is straightforward, only orthogonal structured grids are handled since the alignment of the cell points in each direction is required to derive the appropriate finite differences.

The finite element method is another popular discretization technique, which has been predominantly used in structural problems for the analysis of stress and deformation in solids [35–39]. The discretization technique in the finite element method is based on a variational formulation of the partial differential equation with an error functional to minimize, which provides an approximation of the physical variables, as proved from the calculus of variations. The finite element method requires basis or shape functions defined for reference elements (such as triangles and quadrilaterals), which are then mapped onto the cells. Although the discrete variables are always defined at the vertices, the choice of basis functions provides several derivations of the finite element method. Contrarily to the classical finite difference method, the finite element method does not require a fixed simple mesh structure and, therefore, is capable of handling complex geometries with relative ease, when compared with the former. The linear algebraic equations derived from the method are then assembled in a system of linear equations, which is solved with matrix algebra techniques. However, the derivation of the finite element method requires significant knowledge of calculus of variations and, therefore, is conceptually more complicated than the finite difference method.

The finite volume method is one of the most classical discretization techniques in computational fluid dynamics and has been receiving significant attention since the eighties with the pioneering work of Patankar on heat transfer and fluid flow problems [40]. In comparison with the finite difference method and the finite element method, the extensive use of the finite volume method is recent, although its foundations date back to the early 1970s, previously to the work of Patankar. Posteriorly, the method seems to had been independently used to compute approximations of hyperbolic conservation law systems of compressible gas dynamics in the works of P.W. McDonald, 1971 [41], in the American Society of Mechanical Engineers (ASME) and of R.W. MacCormack et al., 1972 [42], and A.W. Rizzi et al., 1973 [43], in the American Institute of Aeronautics and Astronautics (AIAA). Moreover, the seminal works of B.V. Leer, 1973–1979 [44–49], and the works of V.P. Kolgan, 1972–1975 [50–52], were also remarkable in the early development of the finite volume method for hyperbolic conservation problems. However, some authors claim that the main ideas and principles of the finite volume method are even older and date back to the early 1960s. For instance, the work of S.K. Godunov, 1959 [53], and A. Preissmann, 1961 [54], advocating a box scheme to solve the Saint-Venant equations in hydraulic flow problems, can be regarded as one of the basic finite volume formulations. Additionally, the work of R.S. Varga, 1962 [55], to solve self-adjoint elliptic equations using an integration approach to derive finite difference approximations was standard practice in nuclear research at that time. The works conducted by A.N. Tichnov et al., 1962 [56], and by A.A. Samarskii, 1965 [57], can also be considered as precursors of the finite volume method, although the historical background from the USSR side is scarce. Nevertheless, the method did not receive much attention during those three decades while the finite element method was seeing a noticeable expansion for the physicists and engineers.

1.4.2 Finite volume method

The finite volume method is applied to the integral form of the partial differential equations, usually having a divergence term of some physical variable. The divergence or Gauss's theorem is fundamental in the finite volume method, allowing to convert the volume integral over some finite control volume into a surface integral. For instance, consider some control volume denoted as V, with surface denoted as S, and unit normal vector denoted as $\boldsymbol{n}$, as illustrated in Figure 1.14. For the case of the heat conduction, where the temperature is denoted as T, the governing equation is given as

$$\nabla \cdot (-\kappa \nabla T) = f, \tag{1.1}$$

where κ is the thermal conductivity and f is a heat source, corresponding to the rate of heat generation per unit volume. The finite volume method requires the integral of Equation (1.1) over control volume V, given as

$$\int_V \nabla \cdot (-\kappa \nabla T) \mathrm{d}\boldsymbol{x} = \int_V f \mathrm{d}\boldsymbol{x}. \tag{1.2}$$

Then, the divergence theorem is applied to the left-hand side, which transforms the volume integral of the divergence term into the surface integral of the normal temperature derivative, given as

$$\int_S -\kappa \nabla T \cdot \boldsymbol{n} \mathrm{d}\boldsymbol{x} = \int_V f \mathrm{d}\boldsymbol{x}. \tag{1.3}$$

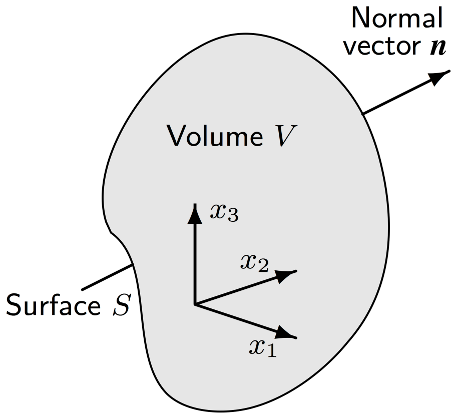
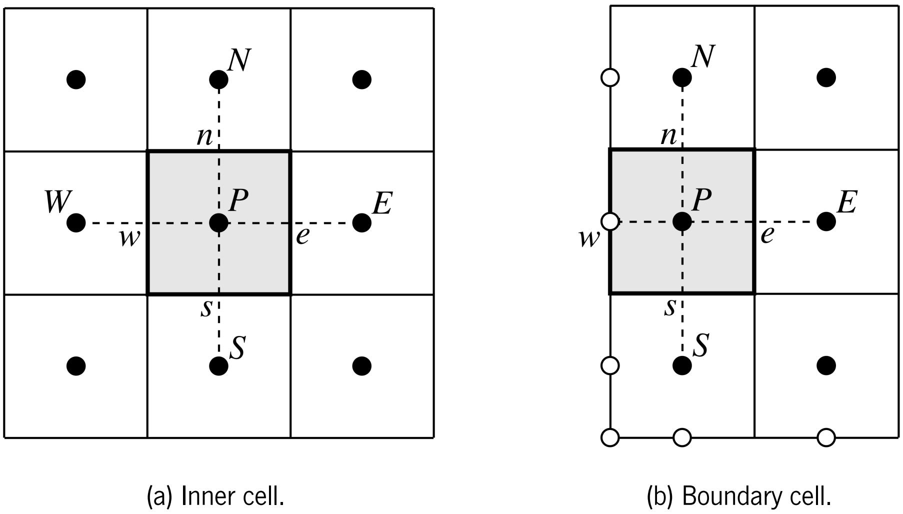


Figure 1.14: Control volume for the finite volume method (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

In practice, the finite volume method consists in applying the same procedure considering the cells of the mesh as the control volumes. For instance, consider a structured mesh with some inner cell denoted as c and faces denoted as n, e, s, and w, as illustrated in Figure 1.15. The volume of the cell is denoted as $|c|$, whereas the areas of the faces are denoted as $|n|$, $|e|$, $|s|$, and $|w|$. The mid-point of the cell is denoted as P, the mid-points of the neighbour cells are noted as N, E, S, and

W, and the outward unit normal vectors denoted as $\boldsymbol{n}_{\mathrm{N}}$, $\boldsymbol{n}_{\mathrm{E}}$, $\boldsymbol{n}_{\mathrm{S}}$, and $\boldsymbol{n}_{\mathrm{W}}$. The unknown temperature is represented piece-wisely at the mid-points of the cells, where for the previous points are denoted as T_{P}, T_{N}, T_{E}, T_{S}, and T_{W}, which are unknowns of the discrete problem.



(a) Inner cell.

(b) Boundary cell.

Figure 1.15: Notations for the finite volume method in a structured orthogonal mesh (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

The division of the surface integral of Equation (1.3) into the different cell faces, gives

$$\int_n -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{N}} \mathrm{d}\boldsymbol{x} + \int_e -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{E}} \mathrm{d}\boldsymbol{x} + \int_s -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{S}} \mathrm{d}\boldsymbol{x} + \int_w -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{W}} \mathrm{d}\boldsymbol{x} = \int_c f \mathrm{d}\boldsymbol{x}. \tag{1.4}$$

Equation (1.4) constitutes a generic formulation of the finite volume method, where no approximation were introduced until this point. However, the surface integrals of the normal temperature derivative on the faces and the volume integral of the source term in the cell have to be discretized to built the equivalent system of linear algebraic equations. In that regard, there is an uncountable number of numerical schemes that emerged within the finite volume method, having in common Equation (1.4) for the problem discretization. A simple and straightforward numerical approximation of the surface integrals in Equation (1.4) is given as

$$\int_n -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{PN}} \mathrm{d}\boldsymbol{x} \approx -\kappa \frac{T_{\mathrm{N}} - T_{\mathrm{P}}}{\left|N_y - P_y\right|} |n|, \tag{1.5}$$

$$\int_e -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{PE}} \mathrm{d}\boldsymbol{x} \approx -\kappa \frac{T_{\mathrm{E}} - T_{\mathrm{P}}}{\left|E_x - P_x\right|} |e|, \tag{1.6}$$

$$\int_s -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{PS}} \mathrm{d}\boldsymbol{x} \approx -\kappa \frac{T_{\mathrm{S}} - T_{\mathrm{P}}}{\left|S_y - P_y\right|} |s|, \tag{1.7}$$

$$\int_w -\kappa \nabla T \cdot \boldsymbol{n}_{\mathrm{PW}} \mathrm{d}\boldsymbol{x} \approx -\kappa \frac{T_{\mathrm{W}} - T_{\mathrm{P}}}{\left|W_x - P_x\right|} |w|, \tag{1.8}$$

which is equivalent to a finite difference approximating the normal derivative of the temperature based on the associated discrete variables. In the case of a cell with a boundary face, as illustrated in Figure 1.15, a different approximation is provided considering the boundary condition prescribed on the associated boundary of the domain. Moreover, in the case of structured or unstructured non-orthogonal meshes, as illustrated in Figure 1.16, such a simple scheme requires non-orthogonal corrections to preserve the consistency of the surface integral approximations. A usual strategy consists in fixing the finite difference between the cell mid-points with a correction based on the vertex points, for which accurate and robust numerical techniques for interpolation of the solution at the vertices are available [58]. On the other side, the volume integral of the source terms is straightforwardly approximated with the value of the given function at the cell mid-point, denoted as f_{P}.

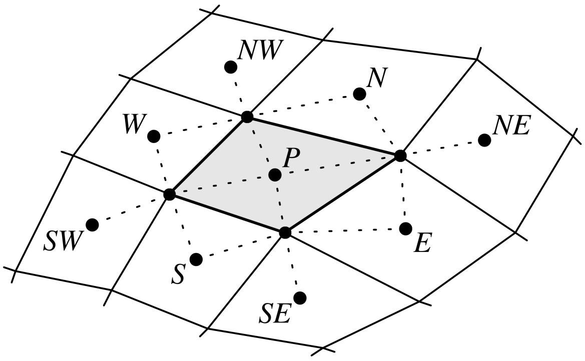


Figure 1.16: Mesh notations for the finite volume method with a structured non-orthogonal mesh (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

Finally, the discrete problem consists in rewritten Equation (1.4) with the numerical approximations of the surface and volume integrals, given as

$$-\kappa\frac{T_{\mathrm{N}}-T_{\mathrm{P}}}{\left|N_y-Py\right|}|n|-\kappa\frac{T_{\mathrm{E}}-T_{\mathrm{P}}}{\left|E_x-P_x\right|}|e|-\kappa\frac{T_{\mathrm{S}}-T_{\mathrm{P}}}{\left|S_y-P_y\right|}|s|-\kappa\frac{T_{\mathrm{W}}-T_{\mathrm{P}}}{\left|W_x-P_x\right|}|w|=f_{\mathrm{P}}|c|, \tag{1.9}$$

which translates into a linear algebraic relation between the discrete variables of the temperature at the cell mid-points. The discretization procedure is repeated for each cell of the mesh, which provides the same number of linear algebraic equations as the number of discrete variables. A system of linear equations is then assembled from these algebraic equations, which is solved using matrix algebra techniques.

In the previous example, the finite volume method has a straightforward physical interpretation since the surface integrals on the control volume surface correspond to the conductive heat flux. Indeed, this translates into the First Law of thermodynamics where the quantity of heat supplied to the control volume has to be equal to the quantity of heat leaving the control volume plus or minus the heat sources or sinks. For other governing equations, by providing the conservation of some quantity in the integral form, the finite volume method leads to the same conservation principle in terms of fluxes of some physical quantity. Indeed, the conservation of specific physical quantities, such as energy and mass, is only mathematically translated when associated partial differential equations are written in the integral form over some control volume. This conservation principle is not only satisfied at the level of the continuous problem but also at the level of the discrete problem based on the balance of the fluxes in the cell. Moreover, the conservation principle in the discrete form is also satisfied regardless of the number or size of cells in the mesh. On the other side, the conservation principle, which is intrinsically satisfied in the finite volume method, is not, however, necessarily verified in the context of the finite difference method or the finite element methods. Another advantage of the finite volume method is the capability to handle any type of mesh, which becomes of interest to work with complex geometries, as often occurs in polymer processing applications. Additionally, the lower abstraction level and higher physical meaning, makes the finite volume method easier to understand and implement, and, thus, more appealing for engineers than the finite element method.

The accuracy of the discretization method depends upon the governing equation, problem geometry, mesh generation, and flux approximation on the control volume surface. Significant developments took place after the work of Patankar for structured meshes and comprehensive literature exists with several classes of methods in the context of the finite volume method, summarized to the following. The classical two-points flux finite volume method [59–62], also referred to as the FV4 method, extends the original Patankar method to unstructured meshes, but an orthogonality condition is required to allow admissible diffusive fluxes. The diamond-cell finite volume method [63–65], was introduced for unstructured non-orthogonal meshes (no orthogonality condition required) and is based on local linear reconstructions to compute the gradients on each control volume face. The drawback of the diamond-cell finite volume method is the lack of symmetry and the difficult and limited numerical analysis. The discrete duality finite volume method [66–68], also referred to as the DDFV method, handles unstructured non-orthogonal and possibly non-conformal meshes and satisfies the div-grad duality intrinsically at the discrete level. Contrarily to the previous methods, the DDFV finite volume method requires unknowns at the vertices of the mesh as well as dual and diamond control volumes in addition to the primal ones. On the other side, the DDFV method is symmetric and numerical analysis is straightforward and general. More recently, to design efficient discretization schemes, several finite volume methods have been proposed, such as the mixed-hybrid methods [69, 70] or the mimetic methods [71, 72], among others.

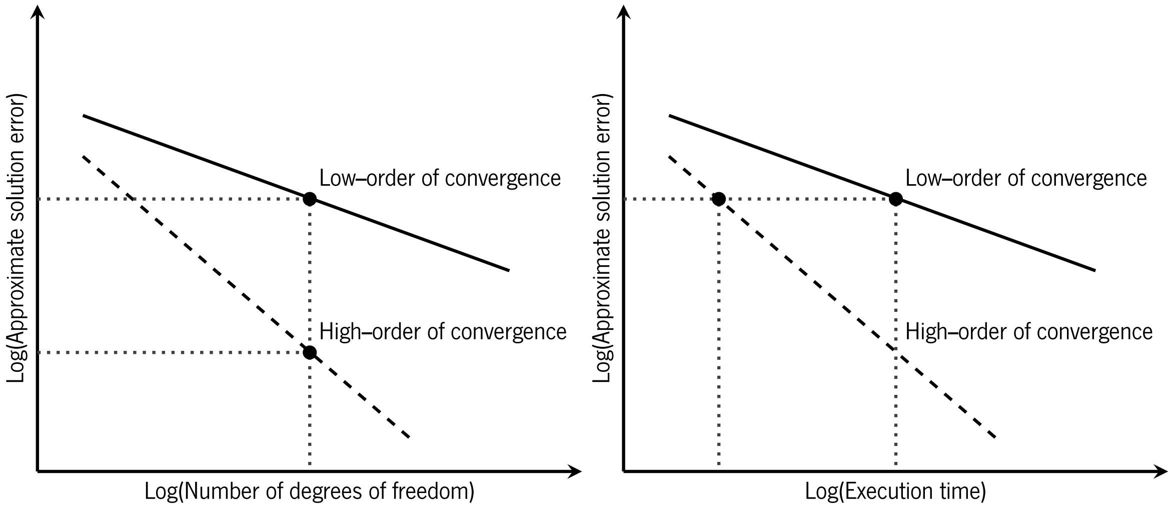
In the case of fluid flow problem, specific numerical techniques are required to correctly solve the div-grad duality between velocity and pressure arising from the balance of momentum and continuity equations [73–79]. In that regard, discretization methods are also classified as staggered, when the discrete variables for the velocity and pressure are defined in different meshes (primal and diamond meshes) [80–82], or as collocated, when both discrete variables are defined in the same mesh [83–85]. Moreover, the discretization methods are also classified as coupled when the governing equations are discretized and solved in the same system of linear equations, leading to a saddle point problem, or as segregated when a projection method in the divergence-free space is used, such as the classical SIMPLE, SIMPLEC, SIMPLER, and PISO methods [86–89].

1.4.3 Very high-order of convergence

The convergence order of the discretization method measures the rate at which the approximate solution error decreases under mesh refinement, that is, the same as increasing the number of unknowns. Classical discretization methods provide a second-order of convergence, whereas achieving higher-orders of convergence usually requires more complex numerical techniques and more laborious implementations. Very high-order accurate methods are herein defined as those having more than the second-order of convergence for the obtained approximate solution error. Such class of methods have historically emerged to capture, with higher accuracy and resolution than the classical ones, shocks and discontinuities in fluid flow hyperbolic-dominated problems, such as the Euler equation for compressible inviscid fluid flows or the shallow water equations. In that context, discretization methods with third and fourth-orders of convergence have been a standard in aeronautic and nuclear research to solve engineering problems with non-smooth solutions. Unfortunately, the computational modelling software commonly used in the polymer processing industry still relies on classical discretization methods with a first- or second-order of convergence.

The most prominent benefit of very high-order accurate methods is to obtain a higher accuracy for the same mesh than low-order accurate methods, such as first- or second-orders, as illustrated in Figure 1.17. Since the approximate solution error decreases faster when employing very high-order accurate methods, this benefit is more pronounced as the accuracy requirements for the application increases. In theory, low-order accurate methods are still capable of achieving the same levels of accuracy increasing the mesh refinement, but with a much higher computational cost and limited in practice by the memory capabilities of computers. Consequently, the memory requirements for demanding applications may exceed the available computer memory, for which very high-order accurate methods are also a promising workaround. From the computational efficiency viewpoint, very high-order accurate methods provide higher accuracy but, as a consequence of their increased complexity, also require more execution time per iteration for the same mesh than low-order accurate methods. In that regard, there is a non-trivial trade-off between accuracy and execution time, which is usually in favour of very high-order accurate methods, as illustrated in Figure 1.17. That is, increasing the convergence order to achieve a certain level of accuracy is usually preferable than refining the mesh with low-order accurate methods, which typically lead to a higher execution time besides the inevitable memory limitations. Notice that time discretization of unsteady problems with very high-order of convergence is also possible, although the literature is more scarce than for the spatial discretization. Nevertheless, these methods are also capable of providing more accurate approximate solutions for long-time simulations, when compared with low-order accurate methods.

In fluid flow hyperbolic-dominated problems, the essentially non-oscillatory method, or its weighted variant, has been widely used to achieve the third- and fifth-orders of convergence [90–95]. The method avoids that spurious oscillations are captured due to the Gibbs phenomenon [96], whereas other approaches have been recently developed, as the multi-dimensional optimal order detection



(a) Number of degrees of freedom.

(b) Execution time.

Figure 1.17: Typical convergence curves for low- and very high-orders accurate methods.

method [97–99]. On the other side, elliptic-dominated problems, as those concerning the heat transfer in polymer processing applications, have not received the same attention within the development of very high-order accurate methods.

CHAPTER 2

Very high-order accurate finite volume scheme for the convection-diffusion equation

Abstract: Obtaining very high-order accurate solutions in curved domains is a challenging task as the accuracy of conventional discretization methods, usually developed for polygonal domains, dramatically reduce without an appropriate treatment of boundary conditions. The classical techniques to preserve the optimal convergence order are found in the context of finite element and finite volume methods, which often rely on curved mesh elements to fit the associated curved boundary. Such techniques demand sophisticated meshing algorithms, cumbersome quadrature rules for integration, and complex non-linear transformations to map the curved mesh elements onto the reference polygonal ones. In this regard, the Reconstruction for Off-site Data method provides very high-order accurate polynomial reconstructions on arbitrary smooth curved boundaries, enabling the integration of the governing equations in polygonal mesh elements and, therefore, avoiding the use of complex integration quadrature rules or non-linear transformations. The method was initially introduced for Dirichlet boundary conditions, and the present work proposes an extension for general boundary conditions, which represents an important advance for real context applications. A generic framework to compute polynomial reconstructions is also developed based on the least-squares method, which handles general constraints and further improves the algorithm. The proposed methods are applied to solve the convection-diffusion equation with a finite volume discretization in general unstructured meshes. A comprehensive numerical benchmark is provided to verify and assess the accuracy, convergence orders, robustness, and efficiency of the method. The results prove that the method is capable of fulfilling boundary conditions appropriately on arbitrary smooth curved boundaries, and a very high-order of convergence is effectively achieved.

Keywords: Very high-order accurate finite volume scheme, arbitrary smooth curved boundaries, general boundary conditions, polynomial reconstructions, least-squares method, reconstruction for off-site data method, convection-diffusion equation

This chapter was adapted from R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, and R. Loubère, Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved domains, Int. J. Numer. Meth. Engrg. 117(2) (2019) 188–220, DOI: https://doi.org/10.1002/nme.5953

2.1 Introduction

The treatment of ever-increasing complexity problems in engineering has only been possible due to the capabilities of modern-day computational methods and high-performance computers. In what concerns to the numerical modelling, computational efficiency is usually determined based on the computational effort necessary to obtain a certain level of solution accuracy, leading to a trade-off between the convergence order of the method and the mesh characteristic size. In the absence of shocks or irregularities, increasing the convergence order is more efficient concerning computational resources than mesh refinement. However, obtaining very high-order accurate approximations is still a challenging task, and many developments in that field are to be made.

The majority of the very high-order accurate methods (more than the second-order) are specifically designed for polygonal (or polyhedral) domains and, usually, numerical difficulties in obtaining the optimal convergence order arise when handling boundary conditions prescribed on curved boundaries. For a short literature review on the topic, the reader is referred to the introduction section in R. Costa et al., 2018 [100], which is summarized in the following. The classical approach to handle boundary conditions on curved boundaries is based on the isoparametric element method [101, 102], which requires, on one side, the introduction of curved mesh elements and, on the other side, non-linear transformations to map the local curved mesh elements onto the reference polygonal ones. An alternative approach, dedicated to the finite volume method, was initially proposed by C. Ollivier-Gooch et al., 2002 [103]. The technique does not require non-linear transformations, but the main shortcoming remains, in particular, the meshing algorithm to generate curved mesh elements fitting curved boundaries in addition to the high-order accurate quadrature rules for numerical integration on non-polygonal mesh elements. As a consequence, handling arbitrary two- or three-dimensional curved elements turns out to be a cumbersome task, which results in significant computational costs [104–106].

R. Costa et al., 2018 [100], introduced a new approach in the finite volume context, the reconstruction for off-site data method (shortened to ROD method), which is capable of handling boundary conditions on arbitrary smooth curved boundaries with a very high-order of convergence. The novelty of the method is to use only polygonal mesh elements, overcoming the mismatch between the mesh boundary and the domain boundary. The method enforces the prescribed boundary conditions using polynomial reconstructions in the vicinity of the boundary, which are computed based on the constrained least-squares method. Moreover, the governing equations are integrated on polygonal mesh elements and, consequently, the numerical heat fluxes are determined solely on the boundaries of the polygonal cells. Therefore, no sophisticated meshing algorithms for curved mesh elements are required, nor non-linear transformations, nor cumbersome quadrature rules for integration in the curved elements. There are very few methods capable of handling curved domains with polygonal meshes, and most of them are limited to the first- or second-order of convergence. Recently an extension of the immersed boundary method to the fourth-order of convergence has been proposed in the framework of the Fourier spectral method [107, 108], which is able of handling arbitrary smooth curved domains.

The ROD method was initially developed only for the steady-state two-dimensional convection-diffusion problem with Dirichlet boundary conditions. In the present work, essential developments are introduced to the method, namely, the handling of Neumann and Robin boundary conditions, which represents a fundamental advance for real context applications. Moreover, the development of a generic framework to compute polynomial reconstructions based on the least-squares method allows the handling of general constraints and improves the algorithm.

The remaining sections of the chapter are organized as follows. Section 2 presents the model, the mesh, and the basic assumptions and notations. Section 3 introduces the generic framework to compute polynomial reconstructions based on the least-squares method. Section 4 is dedicated to the ROD method based on the previously introduced polynomial reconstructions and the Dirichlet, Neumann, and Robin boundary conditions on curved boundaries are addressed. Section 5 presents the very high-order accurate finite volume scheme based on the polynomial reconstructions and the ROD method. Section 6 provides a comprehensive numerical benchmark test suite to verify and assess the proposed method. The chapter is completed in Section 7 with the conclusions and some perspectives for future work.

2.2 Mathematical formulation

The steady-state convection-diffusion problem is addressed in two dimensions and formulated with the Cartesian coordinate system considering $\boldsymbol{x} := (x, y)$. Let Ω be an open bounded domain of $\mathbb{R}^2$ with boundary Γ, partitioned into three non-overlapping and possibly empty subsets, Γ^{D}, Γ^{N}, and Γ^{R}, such that $\Gamma = \Gamma^{\mathrm{D}} \cup \Gamma^{\mathrm{N}} \cup \Gamma^{\mathrm{R}}$. The boundary and the interface are regular Jordan curves, that is, simple and closed curves, and admit a known local parameterization. Vector $\boldsymbol{n} = \big(n_x, n_y\big) := \big(n_x(\boldsymbol{x}), n_y(\boldsymbol{x})\big)$ stands for the outward unit normal vector to Γ at point $\boldsymbol{x}$ on the boundary (see Figure 2.1).

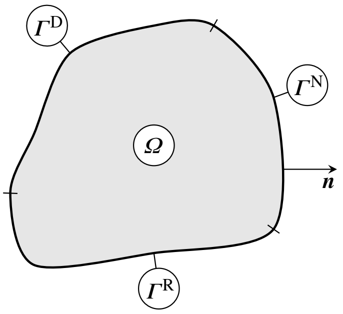


Figure 2.1: Example of arbitrary curved domain with boundary subsets and outward unit normal vector.

2.2.1 Convection-diffusion model

The governing equation for temperature function $\phi := \phi(\boldsymbol{x})$ is given as

$$\nabla \cdot (\boldsymbol{u}\phi - \kappa \nabla \phi) = f, \quad \text{in } \mathit{\Omega}, \tag{2.1}$$

where $\boldsymbol{u} := \left(u_x, u_y\right) := \left(u_x(\boldsymbol{x}), u_y(\boldsymbol{x})\right)$ is the velocity vector function multiplied by the heat capacity and density of the associated material, $\kappa := \kappa(\boldsymbol{x})$ is the thermal conductivity function, and $f := f(\boldsymbol{x})$ is the heat source function (a negative value implies a heat sink). All functions are assumed to be regular and bounded in the domain. To complete Equation (2.1), boundary subsets $\mathit{\Gamma}^{\mathrm{D}}$, $\mathit{\Gamma}^{\mathrm{N}}$, and $\mathit{\Gamma}^{\mathrm{R}}$ are prescribed with the following boundary conditions:

- On boundary subset $\mathit{\Gamma}^{\mathrm{D}}$, a Dirichlet boundary condition is prescribed, given as

  $$\phi = g^{\mathrm{D}}, \quad \text{on } \mathit{\Gamma}^{\mathrm{D}}, \tag{2.2}$$

  where $g^{\mathrm{D}} := g^{\mathrm{D}}(\boldsymbol{x})$ is a given regular function.

- On boundary subset $\mathit{\Gamma}^{\mathrm{N}}$, a Neumann boundary condition is prescribed, given as

  $$-\kappa \nabla \phi \cdot \boldsymbol{n} = g^{\mathrm{N}}, \quad \text{on } \mathit{\Gamma}^{\mathrm{N}}, \tag{2.3}$$

  where $g^{\mathrm{N}} := g^{\mathrm{N}}(\boldsymbol{x})$ is a given regular function.

- On boundary subset $\mathit{\Gamma}^{\mathrm{R}}$, a Robin boundary condition is prescribed, given as

  $$\alpha^{\mathrm{R}}\phi + \beta^{\mathrm{R}} \nabla \phi \cdot \boldsymbol{n} = g^{\mathrm{R}}, \quad \text{on } \mathit{\Gamma}^{\mathrm{R}}, \tag{2.4}$$

  where $g^{\mathrm{R}} := g^{\mathrm{R}}(\boldsymbol{x})$, $\alpha^{\mathrm{R}} := \alpha^{\mathrm{R}}(\boldsymbol{x})$, and $\beta^{\mathrm{R}} := \beta^{\mathrm{R}}(\boldsymbol{x})$ are given regular functions. For instance, if $\alpha^{\mathrm{R}}(\boldsymbol{x}) := \boldsymbol{u}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x})$ and $\beta^{\mathrm{R}}(\boldsymbol{x}) := -\kappa(\boldsymbol{x})$, then the Robin boundary condition represents a total heat flux boundary condition. The Robin boundary condition can also be used to prescribe mixed boundary conditions of Dirichlet and Neumann types.

2.2.2 Polygonal meshes

A general polygonal mesh denoted as $\mathcal{M}$ discretizes the subdomain $\mathit{\Omega}$ and consists of n non-overlapping convex polygonal cells (triangles, quadrangles, etc.). Cells are denoted as c_i with $i \in \mathcal{I} = \{1, \dots, n\}$. Inner edges are denoted as e_{ij} with $j \neq i$ and $i, j \in \mathcal{I}$ and correspond to the edges shared between neighbour cells c_i and c_j and, therefore, $e_{ij} = c_i \cap c_j$. Boundary edges are denoted as e_{iF} with $i \in \mathcal{I}^F$, $F \in \{\mathrm{D}, \mathrm{N}, \mathrm{R}\}$, and correspond to the edges of cells c_i approximating boundary subsets $\mathit{\Gamma}^F$ (for the sake of simplicity, each cell has at most one boundary edge). Subset $\mathcal{I}^F \subset \mathcal{I}$

gathers the indices and n^F is the number of the cells with a boundary edge approximating boundary subset $\mathit{\Gamma}^F$. The vertices of the boundary edges fall on the curves of the associated boundary subsets.

Table 2.1 introduces the geometric properties for the cells and edges and Figures 2.2 provides a schematic representation. Notice that inner edge e_{ij} is also denoted as e_{ji} and, therefore, reference and quadrature points are the same, that is, $\boldsymbol{m}_{ij} = \boldsymbol{m}_{ji}$ and $\boldsymbol{q}_{ij,r} = \boldsymbol{q}_{ji,r}$, whereas outward unit normal vectors are antisymmetric, that is, $\boldsymbol{s}_{ij} = -\boldsymbol{s}_{ji}$.

Table 2.1: Notation and geometric properties for the cells and edges.

Mesh elements	Notation	Properties	Definition	Choice
Cells	c_i	∂c_i	Boundary	
		$\lvert c_i\rvert$	Area	
		$\boldsymbol{m}_i = (m_{i,x}, m_{i,y})$	Reference point (can be any point in c_i)	Centroid
		$\boldsymbol{q}_{i,q} = (q_{i,q,x}, q_{i,q,y})$	Quadrature points, $q = 1,\ldots,Q$	Gaussian
		$\mathcal{N}_i$	Indices of the adjacent cells and boundary subset	
Inner edges	e_{ij}	$\lvert e_{ij}\rvert$	Length	
		$\boldsymbol{m}_{ij} = (m_{ij,x}, m_{ij,y})$	Reference point (can be any point on e_{ij})	Midpoint
		$\boldsymbol{q}_{ij,r} = (q_{ij,r,x}, q_{ij,r,y})$	Quadrature points, $r = 1,\ldots,R$	Gaussian
		$\boldsymbol{s}_{ij} = (s_{ij,x}, s_{ij,y})$	Outward unit normal vector from cell c_i to cell c_j	
Boundary edges	e_{iF}	$\lvert e_{iF}\rvert$	Length	
		$\boldsymbol{m}_{iF} = (m_{iF,x}, m_{iF,y})$	Reference point (can be any point on e_{iF})	Midpoint
		$\boldsymbol{q}_{iF,r} = (q_{iF,r,x}, q_{iF,r,y})$	Quadrature points, $r = 1,\ldots,R$	Gaussian
		$\boldsymbol{s}_{iF} = (s_{iF,x}, s_{iF,y})$	Outward unit normal vector from c_i	

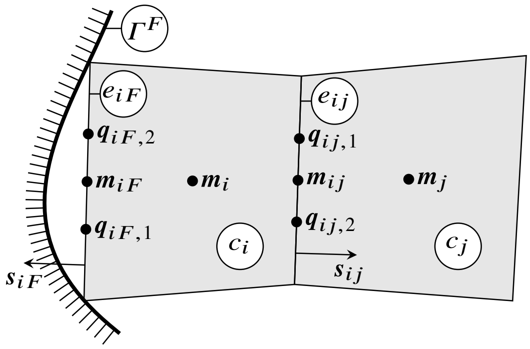


Figure 2.2: Notation and geometric properties for the cells and edges.

In the scope of work, keywords physical and computational distinguish the real domain from the discretized domain, respectively, and in this way, the following definitions are introduced:

- The computational domain, denoted as $\mathit{\Omega}_\Delta$, gathers all the cells and stands for a representative approximation of physical domain Ω, given as

$$\Omega_\Delta = \bigcup_{i \in \mathcal{I}} c_i. \tag{2.5}$$

- The computational boundary, denoted as Γ_Δ, gathers all the boundary edges and stands for a representative approximation of physical boundary Γ, given as

$$\Gamma_\Delta = \bigcup_{i \in \mathcal{I}^F, F \in \{\mathrm{D,N,R}\}} e_{iF}. \tag{2.6}$$

- The computational boundary subset, denoted as Γ_Δ^F, $F \in \{\mathrm{D}, \mathrm{N}, \mathrm{R}\}$, gathers all the boundary edges associated to a boundary subset and stands for a representative approximation of physical boundary subset Γ^F, given as

$$\Gamma_\Delta^F = \bigcup_{i \in \mathcal{I}^F} e_{iF}. \tag{2.7}$$

Remark 1 The curved physical domain, Ω, and corresponding polygonal approximation, Ω_Δ, do not fully overlap as only polygonal meshes are considered. For fine enough meshes, a mismatch of order $\mathcal{O}(h^2)$ is then expected between the physical and the computational boundaries, where h is the characteristic mesh size. Such mismatch represents a potential accuracy deterioration for any more than second-order accurate scheme.

References

[1] R.O. Ebewele, Polymer science and technology, CRC press (2000).

[2] N.G. McCrum, C.P. Buckley, C.B. Bucknall, Principles of polymer engineering, Oxford University Press, New York (1997).

[3] D.M. Bryce, Plastic injection molding: manufacturing process fundamentals, Vol. I: Fundamentals of injection molding series, Society of Manufacturing Engineers, Michigan (1996).

[4] D.M. Bryce, Plastic injection molding: material selection and product design fundamentals, Vol. II: Fundamentals of injection molding series, Society of Manufacturing Engineers, Michigan (1997).

[5] D.M. Bryce, Plastic injection molding: mold design and construction fundamentals, Vol. III: Fundamentals of injection molding series, Society of Manufacturing Engineers, Michigan (1998).

[6] D.M. Bryce, Plastic injection molding: manufacturing startup and management, Vol. IV: Fundamentals of injection molding series, Society of Manufacturing Engineers, Michigan (1999).

[7] T.A. Osswald, J.P. Hernández-Ortiz, Polymer processing: modeling and simulation, Hanser, Munich (2006).

[8] Z. Tadmor, C.G. Gogos, Principles of polymer processing, John Wiley & Sons, New Jersey (2006).

[9] S. Thomas, Y. Weimin (Eds.), Advances in polymer processing: from macro- to nano-scales, Woodhead Publishing Limited, Oxford (2009).

[10] O.S. Carneiro, J.M. Nóbrega, Design of extrusion forming tools, Smithers Rapra, Shawbury (2012).

[11] D.G. Baird, D.I. Collias, Polymer processing: principles and design, 2nd edition, John Wiley & Sons, New Jersey (2014).

[12] A. Rajkumar, Improved methodologies for the design of extrusion forming tools, PhD Thesis, Universidade do Minho (2017).

[13] P. Lin, Y. Jaluria, Conjugate thermal transport in the channel of an extruder for non-Newtonian fluids, Int. J. Heat Mass Transf. 41(21) (1998) 3239–3253.

[14] J. Nizami, Stability analysis and controller design for polymer sheet extrusion, J. Vib. Control 6 (2000) 1083–1105.

[15] D.E. Smith, Design sensitivity analysis and optimization for polymer sheet extrusion and mold filling processes, Int. J. Numer. Meth. Engng. 57 (2003) 1381–1411.

[16] J.M. Nóbrega, O.S. Carneiro, J.A. Covas, F.T. Pinho, P.J. Oliveira, Design of calibrators for extruded profiles. Part I: Modeling the thermal interchanges, Polym. Engrg. Sci. 44(12) (2004) 2216–2228.

[17] J.M. Nóbrega, O.S. Carneiro, Optimising cooling performance of calibrators for extruded profiles, Plast. Rubber Compos. 35(9) (2006) 387–392.

[18] J.M. Nóbrega, O.S. Carneiro, A. Gaspar-Cunha, N.D. Gonçalves, Design of calibrators for profile extrusion – optimizing multi-step systems, Int. Polym. Proc. 23(3) (2008) 331–338.

[19] H. Yang, Conjugate thermal simulation for sheet extrusion die, Polym. Engrg. Sci. 54(3) (2014) 682–694.

[20] F. Marques, S. Clain, G.J. Machado, B. Martins, O.S. Carneiro, J.M. Nóbrega, A novel heat transfer coefficient identification methodology for the profile extrusion calibration stage, Appl. Thermal Engrg 103 (2016) 102–111.

[21] F. Marques, S. Clain, G.J. Machado, B. Martins, O.S. Carneiro, J.M. Nóbrega, A new energy conservation scheme for the numeric study of the heat transfer in profile extrusion calibration, Heat Mass Transf. 53 (2017) 2901–2913.

[22] F. Habla, C. Fernandes, M. Maier, L. Densky, L.L. Ferrás, A. Rajkumar, O.S. Carneiro, O. Hinrichsen, J.M. Nóbrega, Development and validation of a model for the temperature distribution in the extrusion calibration stage, Appl. Therm. Engrg. 100 (2016) 538–552.

[23] M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin (2006).

[24] R.R. Huilgol, N. Phan-Thien, Fluid mechanics of viscoelasticity: general principles, constitutive modelling, analytical and numerical techniques, Rheology Series, Vol. 6, Elsevier Science, Amsterdam (1997).

[25] R.G. Owens, T.N. Phillips, Computational rheology, Vol. 14, Imperial College Press, London (2002).

[26] M.J. Crochet, A.R. Davies, K. Walters, Numerical simulation of non-Newtonian flow, Rheology Series, Vol. 1, Elsevier Science, Amsterdam (2012).

[27] F.A. Morrison, Understanding rheology, Topics in Chemical Engineering, Oxford University Press, New York (2001).

[28] F.P. Beer, E.R. Johnston, J.T. DeWolf, D.F. Mazurek, Mechanics of Materials, McGraw-Hill Education, New York (2014).

[29] R.B. Bird, R.C. Armstrong, O. Hassager, Dynamics of polymeric liquids, Vol. 1: fluid mechanics, 2nd Edition, Wiley-Interscience, New York (1987).

[30] C.W. Macosko, Rheology: principles, measurements, and applications, 1st Edition, Wiley-VCH, New-York (1994).

[31] R.G. Larson, The structure and rheology of complex fluids, Topics in Chemical Engineering, Oxford University Press, New-York (1999).

[32] A. Tveito, R. Winther, Introduction to partial differential equations: a computational approach, Springer-Verlag (1998).

[33] J.W. Thomas, Numerical partial differential equations: finite difference methods, Springer-Verlag (1998).

[34] C. Grossmann, H.-G. Roos, M. Stynes, Numerical treatment of partial differential equations, Springer-Verlag (2007).

[35] J.N. Reddy, An Introduction to the finite element method, McGraw-Hill, New York (1993).

[36] A. Ern, J.-L. Guermond, Theory and practice of finite elements, 159, Springer-Verlag (2004).

[37] Z. Chen, Finite element methods an their applications, Springer-Verlag (2005).

[38] O.C. Zienkiewicz, R.L. Taylor, The finite element method for solid and structural mechanics, Elsevier (2005).

[39] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, The finite element method: its basis and fundamentals, Elsevier (2005).

[40] S.V. Patankar, Numerical heat transfer and fluid flow, Series in Computational Methods in Mechanics and Thermal Sciences, 1st Edition, McGraw-Hill (1980).

[41] P.W. McDonald, The computation of transonic flow through two-dimensional gas turbine cascades, ASME Paper 71–GT–89, International Gas Turbine Conference and Products Show, New York (1971).

[42] R.W. MacCormack, A.J. Paullay, Computational efficiency achieved by time splitting of finite difference operators, AIAA Paper 72–154, 10th Aerospace Sciences Meeting, San Diego, California (1972).

[43] A.W. Rizzi, M. Inouye, Time-split finite-volume method for three-dimensional blunt-body flow, AIAA Journal 11(11) (1973) 1478–1485.

[44] B.V. Leer, Towards the ultimate conservative difference scheme I. The quest of monotonicity, in Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, 163–168, Springer, Berlin (1973).

[45] B.V. Leer, Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme, J. Comput. Phys. 14 (1974) 361–370.

[46] B.V. Leer, MUSCL, A new approach to numerical gas dynamics, Computing in Plasma Physics and Astrophysics, Max-Planck-Institut für Plasma Physik, Garchung, Germany (1976).

[47] B.V. Leer, Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow, J. Comput. Phys. 23 (1977) 263–275.

[48] B.V. Leer, Towards the ultimate conservative difference scheme IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.

[49] B.V. Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.

[50] V.P. Kolgan, Application of the principle of minimum derivatives to the construction of difference schemes for computing discontinuous solutions of gas dynamics, (in Russian) Uch. Zap. TsaGI (Scientific Notes of the Central Institute of Aerodynamics, Russia) 3(6) (1972) 68–77.

[51] V.P. Kolgan, Finite difference schemes for the computation of two-dimensional discontinuous solutions of non-stationary gas dynamics, (in Russian) Uch. Zap. TsaGI (Scientific Notes of the Central Institute of Aerodynamics, Russia) 6(1) (1975) 9–14.

[52] V.P. Kolgan, Finite difference schemes for the computation of three-dimensional solutions of gas dynamics and calculation of the flow over a body under an angle of attack, (in Russian) Uch. Zap. TsaGI (Scientific Notes of the Central Institute of Aerodynamics, Russia) 6(2) (1975) 1–6.

[53] S.K. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, (in Russian) Mat. Sb. 47(89)(3) (1959) 271–306.

[54] A. Preissmann, Propagation des intumescences dans les canaux et le rivières, in Premiere Congrès de l'Association Française de Calcul, Grenoble, France (1961) 433–442.

[55] R.S. Varga, Matrix iterative analysis, 1st Edition, Prentice-Hall: London (1962).

[56] A.N. Tichnov, A.A. Samarskii, Homogeneous difference schemes on non-uniform nets, (in Russian) Zh. Vychisl. Mat. i Mat. Fiz. 2(5) (1962) 812–832.

[57] A.A. Samarskii, Monotonic difference schemes for elliptic and parabolic equations in the case of a non-selfadjoint elliptic operator, (in Russian) Zh. Vychisl. Mat. i Mat. Fiz. 5(3) (1965) 548–551.

[58] R. Costa, S. Clain, G.J. Machado, New cell-vertex reconstruction for finite volume scheme: Application to the convection-diffusion-reaction equation, Comput. Math. Appl. 68(10) (2014) 1229–1249.

[59] Z. Cai, On the finite volume element method, Numer. Math. 58 (1991) 713–735.

[60] Z. Cai, J. Mandel, S. McCormick, The finite volume element method for diffusion equations on general triangulations, SIAM J. Numer. Anal. 28(2) (1991) 392–402.

[61] R. Eymard, T. Gallouët, R. Herbin, The finite volume method, in Ph. Ciarlet, J.L. Lions (Eds.), Handbook for Numerical Analysis, North Holland (2000) 715–1022.

[62] R. Eymard, T. Gallouët, R. Herbin, Finite volume approximation of elliptic problems and convergence of an approximate gradient, Appl. Numer. Math. 37 (2001) 31–53.

[63] Y. Coudière, J.P. Vila, P. Villedieu, Convergence rate of a finite volume scheme for a two dimensional convection diffusion problem, Modél. Math. Anal. Numér. 33(3) (1999) 493–516.

[64] Y. Coudière, P. Villedieu, Convergence rate of a finite volume scheme for the linear convection-diffusion equation on locally refined meshes, M2AN Math, Model. Numer. Anal. 34(6) (1999) 1123–1149.

[65] G. Manzini, A. Russo, A finite volume method for advection-diffusion problems in convection-dominated regimes, Comput. Methods Appl. Mech. Eng. 197 (2008) 1242–1261.

[66] F. Hermeline, A finite volume method for the approximation of diffusion operators on distorted meshes, J. Comput. Phys. 160 (2000) 481–499.

[67] K. Domelevo, P. Omnes, A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids, M2AN Math, Model. Numer. Anal. 39 (2005) 1203–1249.

[68] Y. Coudière, G. Manzini, The discrete duality finite volume method for convection-diffusion problems, SIAM J. Numer. Anal. 47(6) (2010) 4163–4192.

[69] J. Droniou, R. Eymard, A mixed finite volume scheme for anisotropic diffusion problems on any grid, Numer. Math. 105 (2006) 35–71.

[70] R. Eymard, T. Gallouët, R. Herbin, Benchmark on anisotropic problems. SUSHI: a scheme using stabilization and hybrid interfaces for anisotropic heterogeneous diffusion problems, FVCA5- Finite Volumes for Complex Applications V, Wiley (2008) 801–814.

[71] F. Brezzi, K. Lipnikov, M. Shashkov, Convergence of mimetic finite difference methods for diffusion problems on polyhedral meshes, SIAM J. Numer. Anal. 43 (2005) 1872–1896.

[72] A. Cangiani, G. Manzini, Flux reconstruction and solution post-processing in mimetic finite difference methods, Comput. Methods Appl. Mech. Eng. 197(9–12) (2008) 933–945.

[73] A.J. Chorin, Numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 212 (1967) 12–26.

[74] J.H. Ferziger, M. Perić, Computational methods for fluids dynamics, Springer-Verlag, Berlin (1996).

[75] O. Pironneau, Finite element methods for fluids, John Wiley, Chichester (1983).

[76] M.J. Crochet, A.R. Davies, K. Walters, Numerical simulation of non-Newtonian flow, Rheology Series, Vol. 1, Elsevier Science, Amsterdam (1984).

[77] R. Peyret, T.D. Taylor, Computational methods for fluid flow, Springer-Verlag, New York (1985).

[78] R. Temam, Navier-Stokes equations, theory and numerical analysis, North-Holland, Amsterdam (1987).

[79] O.C. Zienkiewicz, R.L. Taylor, P. Nithiarasu, The finite element method for fluid dynamics, Butterworth-Heinemann, Waltham (2014).

[80] M. Piller, E. Stalio, Finite volume compact schemes on staggered grids, J. Comput. Phys. 197 (2004) 299–340.

[81] D. Vidović, A. Segal, P. Wesseling, A superlinearly convergent finite volume method for the incompressible Navier-Stokes equations on staggered unstructured grids, J. Comput. Phys. 198 (2004) 159–177.

[82] N.A. Kampanis, J.A. Ekaterinaris, A staggered grid, high-order accurate method for the incompressible Navier-Stokes Equations, J. Comput. Phys. 215 (2006) 589–613.

[83] P.J. Oliveira, F.T. Pinho, G.A. Pinto, Numerical simulation of non-linear elastic flows with a general collocated finite-volume method, J. Non-Newt. Fluid Mech. 79 (1998) 1–43.

[84] R. Eymard, J.C. Latché, R. Herbin, B. Piar, Convergence of a locally stabilized collocated finite volume scheme for incompressible flows, M2AN 43 (2009) 889–927.

[85] S. Shang, X. Zhao, S. Bayyuk, Generalized formulations for the Rhie-Chow interpolation, J. Comput. Phys. 258 (2014) 880–914.

[86] D. L. Brown, R. Cortez, M. L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 168 (2001) 464–499.

[87] J.L. Guermont, P. Minev, J. Shen, An overview of the projection methods for incompressible flows, Comput. Meth. Appl. Eng. 195 (2006) 6011–6045.

[88] W. Gao, Y.-L. Duan, R.-X. Liu, The finite volume projection method with hybrid unstructured triangular collocated grids for incompressible flows, J. Hydrodyn. 21 (2009) 201–211.

[89] B.E. Griffith, An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as preconditioner, J. Comput. Phys. 228 (2009) 7565–7595.

[90] C. Ollivier-Gooch, High-order ENO schemes for unstructured meshes based on least-squares reconstruction, AIAA Paper 97–0540 (1997).

[91] J.A. Hernaández, High-order finite volume schemes for the advection-diffusion equation, Int. J. Numer. Methods Eng. 53 (2002) 1211–1234.

[92] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, J. Comput. Phys. Arch. 181(2) (2002) 729–752.

[93] E.F. Toro, A. Hidalgo, ADER finite volume schemes for nonlinear reaction-diffusion equations, Appl. Numer. Math. Arch. 59(1) (2009) 1–31.

[94] L. Ivan, C.P.T. Groth, High-order solution-adaptative central essentially non-oscillatory (CENO) method for viscous flows, AIAA 2011–367 (2011).

[95] T. Barth, R. Herbin, M. Ohlberger, Finite volume methods: foundation and analysis, in Encyclopedia of Computational Mechanics (eds. E. Stein, R. Borst, T.J.R. Hughes), 2nd Edition (2017) 1–60.

[96] D. Gottlieb, C.-W.Shu, On the Gibbs phenomenon and its resolution, SIAM Review 39 (1997) 644–668.

[97] S. Clain, S. Diot, R. Loubère, A high-order finite volume method for systems of conservation laws – multi-dimensional optimal order detection (MOOD), J. Comput. Phys. 230 (2011) 4028–4050.

[98] S. Diot, S. Clain, R. Loubère, Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials, Comput. Fluids 64 (2012) 43–63.

[99] S. Diot, R. Loubère, S. Clain, The multidimensional optimal order detection method in the three-dimensional case: very high-order finite volume method for hyperbolic systems, Int. J. Numer. Meth. Fluids 73 (2013) 362–392.

[100] R. Costa, S. Clain, R. Loubère, G.J. Machado, Very high-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state convection-diffusion equation with Dirichlet condition, Appl. Math. Model. 54 (2018) 752–767.

[101] C. Lehrenfeld, High order unfitted finite element methods on level set domains using isoparametric mappings, Comput. Meth. Appl. Mech. Engrg. 300 (2016) 716–733.

[102] C. Lehrenfeld, A. Reusken, Analysis of a high order unfitted finite element method for elliptic interface problems, IMA J. Numer. Anal. 38(3) (2017) 1351–1387.

[103] C. Ollivier-Gooch C, M. Van Altena, A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, J. Comput. Phys. 181(2) (2002) 729–752.

[104] C. Geuzaine, J.F. Remacle, Gmsh: a three-dimensional Finite Element mesh generator with built-in pre- and post-processing facilities, Int. J. Numer. Meth. Engrg. 79 (2009) 1309–1331.

[105] Z.J. Wang, High-order computational fluid dynamics tools for aircraft design, Phil. Trans. R. Soc. A 372 (2014) 20130318.

[106] D. Moxey, M.D. Green, S.J. Sherwin, J. Peiró, An isoparametric approach to high-order curvilinear boundary-layer meshing, Comput. Meth. Appl. Mech. Engrg 283 (2015) 636–650.

[107] D.B. Stein, R.D. Guy, B. Thomases, Immersed boundary smooth extension: A high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods, J. Comput. Phys. 304 (2016) 252–274.

[108] D.B. Stein, R.D. Guy, B. Thomases, Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains, J. Comput. Phys. 335 (2017) 155–178.

Appendix A Least-squares method

The polynomial reconstruction method requires seeking, through a minimization procedure, the polynomial coefficients vector that provides the best approximation of the polynomial function to the cell mean-values chosen according to the associated stencil. For that purpose, the weighted cost functional is defined and given in matrix form as

$$F(\boldsymbol{\eta}) = \left\|\boldsymbol{W}\boldsymbol{A}\boldsymbol{\eta} - \boldsymbol{W}\boldsymbol{b}\right\|_2^2, \tag{A.1}$$

where vector $\boldsymbol{\eta} \in \mathbb{R}^n$ gathers the polynomial coefficients, matrix $\boldsymbol{A} \in \mathbb{R}^{s\times n}$ gathers the coefficients resulting from the evaluation of the monomials mean-values in the cells, diagonal matrix $\boldsymbol{W} \in \mathbb{R}^{s\times s}$ gathers the weights associated to the cells, and vector $\boldsymbol{b} \in \mathbb{R}^s$ gathers the cell mean-values.

Moreover, linear constraint functional $G(\boldsymbol{\eta})$ is required in the case of a constrained polynomial reconstruction to satisfy the cell mean-value conservation or to fulfil the Robin boundary condition. In both cases, constraint functional $G(\boldsymbol{\eta})$ is given in matrix form as

$$G(\boldsymbol{\eta}) = \boldsymbol{c} \cdot \boldsymbol{\eta} - g, \tag{A.2}$$

where vector $\boldsymbol{c} \in \mathbb{R}^n$ gathers the coefficients resulting from the monomials mean-values in the reference cell or from the monomials value at the collocation point and g stands for the cell mean-value to conserve, in the former case, and the point-value to fulfil, in the latter case.

A.1 Normal equations method

For unconstrained polynomial reconstructions, the minimization procedure consists in seeking unique vector $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes weighted cost functional $F(\boldsymbol{\eta})$, that is, $\widetilde{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} F(\boldsymbol{\eta})$. Following the normal equations method, vector $\widetilde{\boldsymbol{\eta}}$ is computed as

$$\widetilde{\boldsymbol{\eta}} = (\boldsymbol{W}\boldsymbol{A})^{\dagger}\boldsymbol{W}\boldsymbol{b} = \left((\boldsymbol{W}\boldsymbol{A})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{A}\right)^{-1}(\boldsymbol{W}\boldsymbol{A})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{b}, = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} \tag{A.3}$$

where $(\boldsymbol{W}\boldsymbol{A})^{\dagger}$ corresponds to the Moore-Penrose pseudoinverse of matrix $\boldsymbol{W}\boldsymbol{A}$ (notice that $\boldsymbol{W}^{\mathrm{T}} = \boldsymbol{W}$ since it is a diagonal matrix). Setting $\boldsymbol{M} = (\boldsymbol{W}\boldsymbol{A})^{\dagger}\boldsymbol{W} \in \mathbb{R}^{s\times n}$, polynomial coefficients vector is given as $\widetilde{\boldsymbol{\eta}} = \boldsymbol{M}\boldsymbol{b}$. The existence of a unique solution for the normal equations method is guaranteed if matrix $\boldsymbol{A}$ has full rank. Since $\boldsymbol{A}$ is a Vandermonde matrix, having at least n distinct cells is a sufficient condition to guarantee the full rank, whereas unstructured meshes are also convenient to avoid ill-conditioned matrices. A preconditioning technique may also be applied to the normal equations method to reduce the condition number and provide more stable results.

Applying the normal equations method in the case of unconstrained polynomial reconstruction $\widetilde{\varphi}_{ij}(\boldsymbol{x})$ for each inner face f_{ij} in computational subdomain Ω^S_Δ, associated polynomial coefficients vector $\widetilde{\boldsymbol{\eta}}_{ij}$ can be computed as $\widetilde{\boldsymbol{\eta}}_{ij} = \widetilde{\boldsymbol{M}}_{ij}\boldsymbol{b}_{ij}$ where vector $\boldsymbol{b}_{ij}$ gathers the cell mean-values and matrix $\widetilde{\boldsymbol{M}}_{ij}$ gathers the associated coefficients obtained from the corresponding terms in Equation (A.3).

A.2 Lagrange multipliers method

For constrained polynomial reconstructions, the minimization procedure consists in seeking unique vector $\widehat{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes weighted cost functional $F(\boldsymbol{\eta})$ and exactly fulfils equation $G(\boldsymbol{\eta}) = 0$, that is, $\widehat{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} F(\boldsymbol{\eta})$ subject to $G(\boldsymbol{\eta}) = 0$.

Following the Lagrange multipliers method, consider functional $L(\boldsymbol{\eta}, \lambda)$ given as

$$L(\boldsymbol{\eta}, \lambda) = F(\boldsymbol{\eta}) + \lambda G(\boldsymbol{\eta}), \tag{A.4}$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier (notice that the number of Lagrange multipliers is the same as the number of linear constraints). Then, polynomial coefficients vector $\widehat{\boldsymbol{\eta}}$ is the solution of system of linear equations $\nabla_{\boldsymbol{\eta},\lambda} L(\boldsymbol{\eta}, \lambda) = \boldsymbol{0}$ where differential operator $\nabla_{\boldsymbol{\eta},\lambda}$ takes the derivatives with respect to each polynomial coefficient and with respect to the Lagrange multiplier.

On one side, $\nabla_{\boldsymbol{\eta}} L(\boldsymbol{\eta}, \lambda) = \nabla_{\boldsymbol{\eta}} F(\boldsymbol{\eta}) + \lambda \nabla_{\boldsymbol{\eta}} G(\boldsymbol{\eta})$ and it can be deduced, after some algebra, that the gradients of cost functional $F(\boldsymbol{\eta})$ and constraint functional $G(\boldsymbol{\eta})$ taking the derivatives with respect to each polynomial coefficient are given in matrix form as

$$\nabla_{\boldsymbol{\eta}} F(\boldsymbol{\eta}) = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A}\boldsymbol{\eta} - 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b}, \tag{A.5}$$

$$\nabla_{\boldsymbol{\eta}} G(\boldsymbol{\eta}) = \boldsymbol{c}, \tag{A.6}$$

and, therefore, $\nabla_{\boldsymbol{\eta}} L(\boldsymbol{\eta}, \lambda)$ is rewritten in matrix form as

$$\nabla_{\boldsymbol{\eta}} L(\boldsymbol{\eta}, \lambda) = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A}\boldsymbol{\eta} - 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} + \lambda\boldsymbol{c}. \tag{A.7}$$

On the other side, notice that $\partial L(\boldsymbol{\eta}, \lambda)/\partial\lambda = G(\boldsymbol{\eta})$ and, therefore, is given in matrix form as in Equation (A.2). Therefore, system of linear equations $\nabla_{\boldsymbol{\eta},\lambda} L(\boldsymbol{\eta}, \lambda) = \boldsymbol{0}$ is rewritten gathering gradient $\nabla_{\boldsymbol{\eta}} L(\boldsymbol{\eta}, \lambda)$ and constraint functional $G(\boldsymbol{\eta})$ equal to the null vector, given in matrix form as

$$\begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A} & \boldsymbol{c} \\ \boldsymbol{c}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} \\ g \end{bmatrix}, \tag{A.8}$$

which implies that polynomial coefficients vector $\widehat{\boldsymbol{\eta}}$ and associated Lagrange multiplier $\widehat{\lambda}$ are given as

$$\begin{bmatrix} \widehat{\boldsymbol{\eta}} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A} & \boldsymbol{c} \\ \boldsymbol{c}^{\mathrm{T}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} \\ g \end{bmatrix}. \tag{A.9}$$

As for the normal equations method, the existence of a unique solution for the normal equations method is guaranteed if matrix $\boldsymbol{A}$ has full rank. To solve system of linear equations (A.8), an LDLT factorization (a closely related variant of the conventional Cholesky factorization, also called LLT factorization) is used since the associated coefficient matrix is symmetric by construction. The Cholesky factorization runs roughly two times faster than the LU factorization if optimally implemented and requires half of the memory with sparse matrix storage. In comparison with the conventional Cholesky factorization, the variant LDLT factorization decomposes the coefficients matrix in a diagonal matrix $\boldsymbol{D}$ in addition to the lower triangular matrix $\boldsymbol{L}$ and, therefore, requires the same memory as the former but avoids extracting square roots, which is computationally convenient. Notice that, in the latter, matrix $\boldsymbol{L}$ is a unitriangular matrix (the entries in the diagonal are ones) and, therefore, matrices $\boldsymbol{D}$ and $\boldsymbol{L}$ can be stored in a single lower triangular matrix. Moreover, for indefinite matrices with no conventional Cholesky factorization, the variant LDLT factorization can be computed having negative entries in the diagonal matrix. Another technique to obtain more accurate and stable results is to apply a preconditioning matrix to the Lagrange multipliers method to reduce the condition number of its associated coefficient matrix $\boldsymbol{A}$.

Applying the Lagrange multipliers method in the case of constrained polynomial reconstruction $\widehat{\varphi}_i(\boldsymbol{x})$ for each cell c_i in computational subdomain Ω_{Δ}^{S}, associated polynomial coefficients vector $\widehat{\boldsymbol{\eta}}_i$ can be computed given as $\widehat{\boldsymbol{\eta}}_i = \widehat{\boldsymbol{M}}_i\boldsymbol{b}_i$ where vector $\boldsymbol{b}_i$ gathers the cell mean-values and matrix $\widehat{\boldsymbol{M}}_i$ gathers the associated coefficients obtained from the corresponding terms in Equation (A.9). Notice that, in that case, $g = \phi_i$ is a cell mean-value and, therefore, is also gathered in vector $\boldsymbol{b}_i$.

Applying the Lagrange multipliers method in the case of constrained polynomial reconstruction $\widehat{\varphi}_{iF}(\boldsymbol{x})$ for each boundary face f_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$, associated polynomial coefficients vector $\widehat{\boldsymbol{\eta}}_{iF}$ can be computed given as $\widehat{\boldsymbol{\eta}}_{iF} = \widehat{\boldsymbol{M}}_{iF}\boldsymbol{b}_{iF} + g_{iF}\widehat{\boldsymbol{n}}_{iF}$ where vector $\boldsymbol{b}_{iF}$ gathers the cell mean-values and matrix $\widehat{\boldsymbol{M}}_{iF}$ and vector $\widehat{\boldsymbol{n}}_{iF}$ gather the coefficients associated to the cell mean-values and to the point-value, respectively, obtained from the corresponding terms in Equation (A.9). Notice that, in that case, $g = g_{iF}$ is a point-value and, therefore, is not gathered in vector $\boldsymbol{b}_{iF}$.

Appendix B Principal curvatures and directions

Contrarily to planar curves, the normal curvature of surfaces generally depends on the chosen direction. In that regard, the classical literature in differential geometry often provides curvature formulas for the Gaussian and mean curvatures at any point on the surface. The Gaussian curvature evaluates the rate of deviation between points on the surface (intrinsic property), while the mean curvature is a measure of the rate of deviation between the surface and the tangent plane (extrinsic property). Moreover, at any point on the surface, there are two intersecting lines contained on the surface with the smallest and largest radii of curvature, which are called the principal radii of curvature, while their reciprocals are called the principal curvatures. In that regard, the mean curvature of the surface corresponds to the average of the principal curvatures, while the Gaussian curvature corresponds to the product. Denote as $\kappa_{\max} := \kappa_{\max}(\boldsymbol{x})$ and $\kappa_{\min} := \kappa_{\min}(\boldsymbol{x})$ the principal curvatures and as $\kappa_{\mathrm{G}} := \kappa_{\mathrm{G}}(\boldsymbol{x})$ and $\kappa_{\mathrm{M}} := \kappa_{\mathrm{M}}(\boldsymbol{x})$ the Gaussian and mean curvatures, respectively. Then, from the above definitions

$$\kappa_{\mathrm{G}} = \kappa_{\max}\kappa_{\min} \qquad \text{and} \qquad \kappa_{\mathrm{M}} = \frac{\kappa_{\max}+\kappa_{\min}}{2}, \tag{B.1}$$

which are solved for the principal curvatures, yielding

$$\kappa_{\max} = \kappa_{\mathrm{M}} + \sqrt{(\kappa_{\mathrm{M}})^2 - \kappa_{\mathrm{G}}} \qquad \text{and} \qquad \kappa_{\min} = \kappa_{\mathrm{M}} - \sqrt{(\kappa_{\mathrm{M}})^2 - \kappa_{\mathrm{G}}}. \tag{B.2}$$

The associated directions (referred to as principal directions) are represented with vectors $\boldsymbol{\tau}_{\max} := \boldsymbol{\tau}_{\max}(\boldsymbol{x})$ and $\boldsymbol{\tau}_{\min} := \boldsymbol{\tau}_{\min}(\boldsymbol{x})$, respectively. In that regard, the maximum and minimum principal directions are assigned to the unit tangential and bi-tangential vectors, respectively, hence $\boldsymbol{t}(\boldsymbol{x}) = \boldsymbol{\tau}_{\max}(\boldsymbol{x})/\|\boldsymbol{\tau}_{\max}(\boldsymbol{x})\|$ and $\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{\tau}_{\min}(\boldsymbol{x})/\|\boldsymbol{\tau}_{\min}(\boldsymbol{x})\|$. Notice that $\boldsymbol{\tau}_{\max}(\boldsymbol{x})$ and $\boldsymbol{\tau}_{\min}(\boldsymbol{x})$ are not necessarily unitary vectors. Accordingly, the associated curvatures are $\kappa_t(\boldsymbol{x}) = \kappa_{\max}(\boldsymbol{x})$ and $\kappa_s(\boldsymbol{x}) = \kappa_{\min}(\boldsymbol{x})$.

Since the curvature is a second-order effect, only the first- and second-order partial derivatives of the surface function are usually required to calculate the associated curvature. In that regard, only regular surfaces with, at least, C^2-continuity are considered, that is, the surface must be twice continuously differentiable. Moreover, only analytical representations of the boundary, either parametric or implicit, are addressed in the present work, such that the required derivatives are easily obtained.

B.1 Parametric surfaces

Consider a parametric surface with $\boldsymbol{x} = \boldsymbol{p}(u, v)$, where u and v are two quantities parametrising the surface. Denote as $\boldsymbol{p}_u := \boldsymbol{p}_u(u, v)$ and $\boldsymbol{p}_v := \boldsymbol{p}_v(u, v)$ the first-order partial derivatives of $\boldsymbol{p}(u, v)$ with respect to parameters u and v, respectively, and denote as $\boldsymbol{p}_{uu} := \boldsymbol{p}_{uu}(u, v)$, $\boldsymbol{p}_{uv} := \boldsymbol{p}_{uv}(u, v)$, $\boldsymbol{p}_{vv} := \boldsymbol{p}_{vv}(u, v)$, and $\boldsymbol{p}_{vu} := \boldsymbol{p}_{vu}(u, v)$ the second-order partial derivatives of $\boldsymbol{p}(u, v)$ with respect to u and v, according to the indices. Moreover, consider matrix $\boldsymbol{T} := \boldsymbol{T}(u, v)$ given as

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{p}_u & \boldsymbol{p}_v \end{bmatrix}. \tag{B.3}$$

Vectors $\boldsymbol{p}_u(u, v)$ and $\boldsymbol{p}_v(u, v)$ correspond to two tangential vectors on the surface (not necessarily unitary) and, since parameters u and v are assumed to be arbitrary, are not necessarily in the directions of the principal curvatures. On the other side, the normal unit vector on the surface, $\boldsymbol{n} := \boldsymbol{n}(u, v)$, is easily obtained from these tangential vectors as

$$\boldsymbol{n} = \frac{\boldsymbol{p}_u \times \boldsymbol{p}_v}{\|\boldsymbol{p}_u \times \boldsymbol{p}_v\|}, \tag{B.4}$$

for which $\boldsymbol{n}_u := \boldsymbol{n}_u(u, v)$ and $\boldsymbol{n}_v := \boldsymbol{n}_v(u, v)$ denote the first-order partial derivatives with respect to parameters u and v, respectively. Notice that the orientation of the normal vector is determined from the chosen surface parametrisation and, therefore, the previous formula does not ensure the conventional orientation, that is, outwards the domain.

The classical curvature formulas for parametric surfaces usually depend on the associated first and second fundamental forms, denoted as $\boldsymbol{I} := \boldsymbol{I}(u, v)$ and $\boldsymbol{II} := \boldsymbol{II}(u, v)$, respectively. The first fundamental form, which describes the metric properties of a surface, corresponds to the inner product on the tangent space and is determined from the inner product of the first-order partial derivatives of the surface. On the other side, the second fundamental form (also referred to as shape tensor) corresponds to the surface quadratic form on the tangent plane and is determined from the projections of the second-order partial derivatives of the surface onto the surface normal. Together with the first fundamental form, the second fundamental form serves to define extrinsic invariants of the surface, such as principal curvatures. Based on these definitions, the first and second fundamental forms are more formally given as

$$\boldsymbol{I} = \begin{bmatrix} \boldsymbol{p}_u \cdot \boldsymbol{p}_u & \boldsymbol{p}_u \cdot \boldsymbol{p}_v \\ \boldsymbol{p}_v \cdot \boldsymbol{p}_u & \boldsymbol{p}_v \cdot \boldsymbol{p}_v \end{bmatrix} \quad \text{and} \quad \boldsymbol{II} = \begin{bmatrix} \boldsymbol{p}_{uu} \cdot \boldsymbol{n} & \boldsymbol{p}_{uv} \cdot \boldsymbol{n} \\ \boldsymbol{p}_{vu} \cdot \boldsymbol{n} & \boldsymbol{p}_{vv} \cdot \boldsymbol{n} \end{bmatrix} = -\begin{bmatrix} \boldsymbol{p}_u \cdot \boldsymbol{n}_u & \boldsymbol{p}_u \cdot \boldsymbol{n}_v \\ \boldsymbol{p}_v \cdot \boldsymbol{n}_u & \boldsymbol{p}_v \cdot \boldsymbol{n}_v \end{bmatrix}. \tag{B.5}$$

For the second fundamental form, notice that $\boldsymbol{p}_u(u, v) \cdot \boldsymbol{n}(u, v) = 0$ and, therefore, differentiating this equality with respect to u yields $\boldsymbol{p}_{uu}(u, v) \cdot \boldsymbol{n}(u, v) + \boldsymbol{p}_u(u, v) \cdot \boldsymbol{n}_u(u, v) = 0$, from which it is deduced that $\boldsymbol{p}_{uu}(u, v) \cdot \boldsymbol{n}(u, v) = -\boldsymbol{p}_u(u, v) \cdot \boldsymbol{n}_u(u, v)$. The same procedure is applied for the remaining terms of the second fundamental form to establish the second matrix definition presented in the above equation.

Consider parameters $E := E(u,v)$, $F := F(u,v)$, $G := G(u,v)$, $L := L(u,v)$, $M := M(u,v)$, and $N := N(u,v)$, which are intrinsically related with the definitions of the first and second fundamental forms, and read as

$$\begin{aligned} E &= \boldsymbol{p}_u \cdot \boldsymbol{p}_u, & F &= \boldsymbol{p}_u \cdot \boldsymbol{p}_v = \boldsymbol{p}_v \cdot \boldsymbol{p}_u, & G &= \boldsymbol{p}_v \cdot \boldsymbol{p}_v, \\ L &= \boldsymbol{p}_{uu} \cdot \boldsymbol{n}, & M &= \boldsymbol{p}_{uv} \cdot \boldsymbol{n} = \boldsymbol{p}_{vu} \cdot \boldsymbol{n}, & N &= \boldsymbol{p}_{vv} \cdot \boldsymbol{n}. \end{aligned} \tag{B.6}$$

Moreover, consider the adjugate matrix (or classical adjoint matrix) of the second fundamental form, denoted as $\mathrm{adj}(\boldsymbol{II})$, which corresponds to the cofactors matrix transpose, that is

$$\mathrm{adj}(\boldsymbol{II}) = \begin{bmatrix} \boldsymbol{p}_{vv} \cdot \boldsymbol{n} & -\boldsymbol{p}_{vu} \cdot \boldsymbol{n} \\ -\boldsymbol{p}_{uv} \cdot \boldsymbol{n} & \boldsymbol{p}_{uu} \cdot \boldsymbol{n} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{p}_v \cdot \boldsymbol{n}_v & \boldsymbol{p}_v \cdot \boldsymbol{n}_u \\ \boldsymbol{p}_u \cdot \boldsymbol{n}_v & -\boldsymbol{p}_u \cdot \boldsymbol{n}_u \end{bmatrix}. \tag{B.7}$$

Then, with some algebraic manipulations, several equivalent formulas are derived for the Gaussian and mean curvatures at a given point on the surface, given as

$$\kappa_{\mathrm{G}} = \frac{\det(\boldsymbol{II})}{\det(\boldsymbol{I})} = \frac{LN - M^2}{EG - F^2} = \frac{(\boldsymbol{p}_u \times \boldsymbol{p}_v) \cdot (\boldsymbol{n}_u \times \boldsymbol{n}_v)}{\|\boldsymbol{p}_u \times \boldsymbol{p}_v\|^2}, \tag{B.8}$$

$$\kappa_{\mathrm{M}} = \frac{\mathrm{trace}(\boldsymbol{I}\,\mathrm{adj}(\boldsymbol{II}))}{2\det(\boldsymbol{I})} = \frac{EN + GL - 2FM}{2(EG - F^2)} = \frac{(\boldsymbol{p}_u \times \boldsymbol{p}_v) \cdot ((\boldsymbol{p}_v \times \boldsymbol{n}_u) - (\boldsymbol{p}_u \times \boldsymbol{n}_v))}{2\|\boldsymbol{p}_u \times \boldsymbol{p}_v\|^2}, \tag{B.9}$$

respectively. The principal curvatures are easily obtained from the Gaussian and mean curvatures using the corresponding definitions (B.2). On the other side, the principal directions are determined from the corresponding slopes in the bi-dimensional uv-plane, denoted as $\lambda_{\max} := \lambda_{\max}(u,v)$ and $\lambda_{\min} := \lambda_{\min}(u,v)$, given as

$$\lambda_{\max} = -\frac{M - \kappa_{\max}F}{N - \kappa_{\max}G} = -\frac{L - \kappa_{\max}E}{M - \kappa_{\max}F} \qquad \text{and} \qquad \lambda_{\min} = -\frac{M - \kappa_{\min}F}{N - \kappa_{\min}G} = -\frac{L - \kappa_{\min}E}{M - \kappa_{\min}F}, \tag{B.10}$$

which are represented with vectors $\boldsymbol{v}_{\max} := \boldsymbol{v}_{\max}(u,v) = (1, \lambda_{\max})^{\mathrm{T}}$ and $\boldsymbol{v}_{\min} := \boldsymbol{v}_{\min}(u,v) = (1, \lambda_{\min})^{\mathrm{T}}$, and transformed to the three-dimensional space as $\boldsymbol{\tau}_{\max}(u,v) = \boldsymbol{T}(u,v)\boldsymbol{v}_{\max}(u,v)$ and $\boldsymbol{\tau}_{\min}(u,v) = \boldsymbol{T}(u,v)\boldsymbol{v}_{\min}(u,v)$.

Consider the so-called Weingarten matrix, denoted as $\boldsymbol{W} := \boldsymbol{W}(u,v)$, given as $\boldsymbol{W}(u,v) = \boldsymbol{II}(u,v)\boldsymbol{I}^{-1}(u,v)$ based on the first and second fundamental forms, which admits an eigendecomposition in the form $\boldsymbol{W}(u,v)\boldsymbol{v}(u,v) = \lambda(u,v)\boldsymbol{v}(u,v)$ with eigenvalues $\lambda := \lambda(u,v) \in \{\lambda_1(u,v), \lambda_2(u,v)\}$ and eigenvectors $\boldsymbol{v} := \boldsymbol{v}(u,v) \in \{\boldsymbol{v}_1(u,v), \boldsymbol{v}_2(u,v)\}$. Then, the eigenvalues of the Weingarten matrix correspond to the principal curvatures, hence $\kappa_{\max}(u,v) = \lambda_1(u,v)$ and $\kappa_{\min}(u,v) = \lambda_2(u,v)$ assuming that $\lambda_1(u,v) \geq \lambda_2(u,v)$. On the other side, the associated eigenvectors correspond to the principal directions represented in the bi-dimensional

uv-plane, which are transformed to the three-dimensional space as $\boldsymbol{\tau}_{\max}(u,v) = \boldsymbol{T}(u,v)\boldsymbol{v}_1(u,v)$ and $\boldsymbol{\tau}_{\min}(u,v) = \boldsymbol{T}(u,v)\boldsymbol{v}_2(u,v)$.

Although the latter procedure avoids the calculation of the intermediate Gaussian and mean curvatures, a robust eigendecomposition algorithm is required to prevent loss of precision, particularly at (nearly) umbilical and planar points. Nevertheless, having alternative formulas for the computation of the principal curvatures allows double-checking the correct implementation of these formulas, especially in complex parametrised surfaces.

B.2 Implicit surfaces

Consider an implicit surface given with $\phi(\boldsymbol{x}) = 0$, and denote as $\boldsymbol{G} := \boldsymbol{G}(\boldsymbol{x})$ and $\boldsymbol{H} := \boldsymbol{H}(\boldsymbol{x})$ the gradient vector and Hessian matrix of function $\phi(\boldsymbol{x})$, respectively, given as

$$\boldsymbol{G} = \begin{bmatrix} \dfrac{\partial\phi}{\partial x} & \dfrac{\partial\phi}{\partial y} & \dfrac{\partial\phi}{\partial z} \end{bmatrix}^{\mathrm{T}} \qquad \text{and} \qquad \boldsymbol{H} = \begin{bmatrix} \dfrac{\partial^2\phi}{\partial x^2} & \dfrac{\partial^2\phi}{\partial y^2} & \dfrac{\partial^2\phi}{\partial z^2} \\ \dfrac{\partial^2\phi}{\partial y\partial x} & \dfrac{\partial^2\phi}{\partial y^2} & \dfrac{\partial^2\phi}{\partial y\partial z} \\ \dfrac{\partial^2\phi}{\partial z\partial x} & \dfrac{\partial^2\phi}{\partial z\partial y} & \dfrac{\partial^2\phi}{\partial z^2} \end{bmatrix}. \tag{B.11}$$

The unit normal vector on the surface, denoted as $\boldsymbol{n} := \boldsymbol{n}(\boldsymbol{x})$, can be easily obtained as

$$\boldsymbol{n} = \frac{\boldsymbol{G}}{||\boldsymbol{G}||}. \tag{B.12}$$

As for parametric surfaces, the orientation of the normal vector is determined from the chosen surface equation, and, therefore, the previous formula does not ensure the conventional orientation, that is, outwards the domain. In particular, if $\phi(\boldsymbol{x}) > 0$ for any point outside the domain, then the normal vector provided from the previous formula is oriented outwards.

The literature on the curvature calculation for implicit surfaces is scarcer than for parametric surfaces. Consider the adjugate matrix (or classical adjoint matrix) of the Hessian matrix, denoted as $\operatorname{adj}(\boldsymbol{H})$, which corresponds to the cofactors matrix transpose, that is

$$\operatorname{adj}(\boldsymbol{H}) = \begin{bmatrix} H_{22}H_{33} - H_{23}H_{32} & H_{13}H_{32} - H_{12}H_{33} & H_{12}H_{23} - H_{13}H_{22} \\ H_{23}H_{31} - H_{21}H_{33} & H_{11}H_{33} - H_{13}H_{31} & H_{21}H_{13} - H_{11}H_{23} \\ H_{21}H_{32} - H_{22}H_{31} & H_{12}H_{31} - H_{11}H_{32} & H_{11}H_{22} - H_{12}H_{21} \end{bmatrix}, \tag{B.13}$$

where $H_{ij} := H_{ij}(\boldsymbol{x})$, $i, j \in \{1,2,3\}$, correspond to the second-order partial derivatives in the Hessian matrix. Then, the Gaussian and mean curvatures at a given point on the surface are obtained as

$$\kappa_{\mathrm{G}} = \frac{\boldsymbol{G}\,\mathrm{adj}(\boldsymbol{H})\boldsymbol{G}^{\mathrm{T}}}{\|\boldsymbol{G}\|^4} \qquad \text{and} \qquad \kappa_{\mathrm{M}} = \frac{\boldsymbol{G}\boldsymbol{H}\boldsymbol{G}^{\mathrm{T}} - \|\boldsymbol{G}\|^2\,\mathrm{trace}(\boldsymbol{H})}{2\|\boldsymbol{G}\|^3}. \tag{B.14}$$

The principal curvatures are easily obtained from the Gaussian and mean curvatures using the corresponding definitions (B.2).

In some works, where only the mean curvature is necessary, an alternative formula is often employed based on the divergence of the unit normal vector, given as

$$\kappa_{\mathrm{M}} = -\frac{1}{2}\nabla\cdot\boldsymbol{n} = -\frac{1}{2}\nabla\cdot\left(\frac{\nabla\phi}{\|\nabla\phi\|}\right), \tag{B.15}$$

which is shorter and simpler than the previous formula. Nevertheless, when the Gaussian or principal curvatures are also required, the gradient vector and the Hessian matrix still need to be computed.

Consider two orthonormal tangential vectors on the surface, denoted as $\boldsymbol{\tau}_1 := \boldsymbol{\tau}_1(\boldsymbol{x})$ and $\boldsymbol{\tau}_2 := \boldsymbol{\tau}_2(\boldsymbol{x})$, hence also orthonormal to the normal vector. The choice of these tangential vectors is arbitrary and, therefore, a simple technique consists in choosing a random vector $\boldsymbol{\tau}_0$ and, then, computing vectors $\boldsymbol{\tau}_1(\boldsymbol{x}) = \boldsymbol{n}(\boldsymbol{x})\times\boldsymbol{\tau}_0$ and $\boldsymbol{\tau}_2(\boldsymbol{x}) = \boldsymbol{n}(\boldsymbol{x})\times\boldsymbol{\tau}_1(\boldsymbol{x})$. Such a technique requires that vector $\boldsymbol{\tau}_0$ is non-collinear with the normal vector, for instance, one of the canonical basis vectors. Moreover, consider matrix $\boldsymbol{T} := \boldsymbol{T}(\boldsymbol{x})$ given as

$$\boldsymbol{T} = \begin{bmatrix}\boldsymbol{\tau}_1 & \boldsymbol{\tau}_2\end{bmatrix}, \tag{B.16}$$

and matrix $\boldsymbol{A} := \boldsymbol{A}(\boldsymbol{x})$ given as

$$\boldsymbol{A} = \frac{\boldsymbol{T}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{T}}{\|\boldsymbol{H}\|}, \tag{B.17}$$

which admits an eigendecomposition in the form $\boldsymbol{A}(\boldsymbol{x})\boldsymbol{v}(\boldsymbol{x}) = \lambda(\boldsymbol{x})\boldsymbol{v}(\boldsymbol{x})$ with eigenvalues $\lambda := \lambda(\boldsymbol{x}) \in \{\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x})\}$ and eigenvectors $\boldsymbol{v} := \boldsymbol{v}(\boldsymbol{x}) \in \{\boldsymbol{v}_1(\boldsymbol{x}), \boldsymbol{v}_2(\boldsymbol{x})\}$. Similarly to parametric surfaces, the eigenvalues correspond to the principal curvatures, hence $\kappa_{\max}(\boldsymbol{x}) = \lambda_1(\boldsymbol{x})$ and $\kappa_{\min}(\boldsymbol{x}) = \lambda_2(\boldsymbol{x})$ assuming that $\lambda_1(\boldsymbol{x}) \geq \lambda_2(\boldsymbol{x})$. On the other side, the associated eigenvectors correspond to the principal directions represented in the bi-dimensional space, which are transformed to the three-dimensional space as $\boldsymbol{\tau}_{\max}(\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x})\boldsymbol{v}_1(\boldsymbol{x})$ and $\boldsymbol{\tau}_{\min}(\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x})\boldsymbol{v}_2(\boldsymbol{x})$.

A point $\boldsymbol{p}$ on the surface is called umbilical if its principal curvatures are equal, that is $\kappa_{\max}(\boldsymbol{p}) = \kappa_{\min}(\boldsymbol{p})$, meaning that the surface bends the same amount in all directions and, consequently, every tangent vector at $\boldsymbol{p}$ represents a principal direction. Umbilical points can be regarded as points that locally represent either a plane ($\kappa_{\max}(\boldsymbol{p}) = \kappa_{\min}(\boldsymbol{p}) = 0$) or a spherical surface ($\kappa_{\max}(\boldsymbol{p}) = \kappa_{\min}(\boldsymbol{p}) \neq 0$). Since umbilical points are singularities of the principal directions, a robust implementation of the classical formulas presented above is necessary to appropriately address indeterminate solutions.