Universidade do Minho
Escola de Engenharia

Ricardo Daniel Pereira da Costa

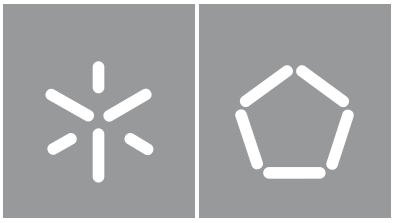
Advanced computational methods towards high-performance polymer processing simulations

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

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).

I am sincerely grateful and respectful to my supervisors, Professors João Miguel Nóbrega and Stéphane Clain, and also to Professor Gaspar Machado, whom I have known and worked with for almost a decade. Your guidance, support, persistence, and outstanding knowledge were absolutely indispensable to undertake and conclude my doctoral thesis, accomplishing great and memorable successes, and for that, I am genuinely thankful. Besides your scientific supervision, I also thank you for the valuable personal and professional advice you gave me, which I always take with much consideration in my life.

I also express my gratitude to Professor João Pedro Nunes, director of the doctoral program in Science and Engineering of Polymers and Composites, for the advice and support provided, especially regarding the administrative processes of the doctoral program.

I am also sincerely thankful to those who fulfilled my days with love, caring, friendship, companionship, and good times and, more importantly, who supported me through the bad times with motivation, strength, and patience. Special gratitude to my parents, sister, grandparents, José Rocha, Cláudia Fernandes, Hugo Dinis, Filipa Ferraz, Ivo Ramalhosa, Adriano Pinto, Pedro Rodrigues, Dalila Vieira, Ana Soares, Cidália Castro, Liliana Melro, Nuno Ramos, Mohammadreza Aali, and João Pedro.

Finally, I sincerely acknowledge the friendship and support of my colleagues, secretariat, professors from the Department of Polymer Engineering, professors from the Department of Mathematics and Applications, and to those at the University of Minho that, somehow, helped me to thrive in this incredible journey.

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: _____

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

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

Publications and awards

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

- 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.
- R. Costa, J.M. Nóbrega, S. Clain, and G.J. Machado, Very high-order accurate polygonal mesh finite volume scheme for conjugate heat transfer problems with curved interfaces and imperfect contacts, Comput. Meth. Appl. Mech. Engrg. 357 (2019) 112560, DOI: https://doi.org/10.1016/j.cma.2019.07.029.
- R. Costa, J.M. Nóbrega, S. Clain, and G.J. Machado, Sheet extrusion cooling stage simulation with a very high-order accurate finite volume scheme, in preparation for Int. J. Heat Mass Tran. (2020).
- R. Costa, J.M. Nóbrega, S. Clain, and G.J. Machado, Efficient very high-order accurate polyhedral mesh finite volume scheme for 3D conjugate heat transfer problems in curved domains, in preparation for J. Comput. Phys. (2020).

The present work was also recognized with the PPS Graduate Travel Award in the 35th International Conference of the Polymer Processing Society, held on May 26–30, 2019, in Çeşme, İzmir, Turkey.

Table of contents

List of figures

List of tables

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

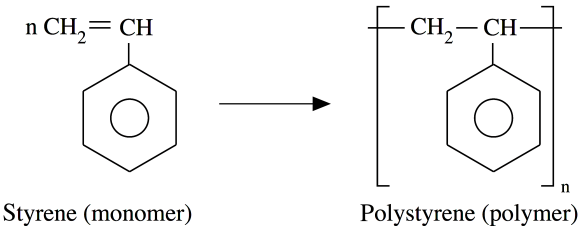
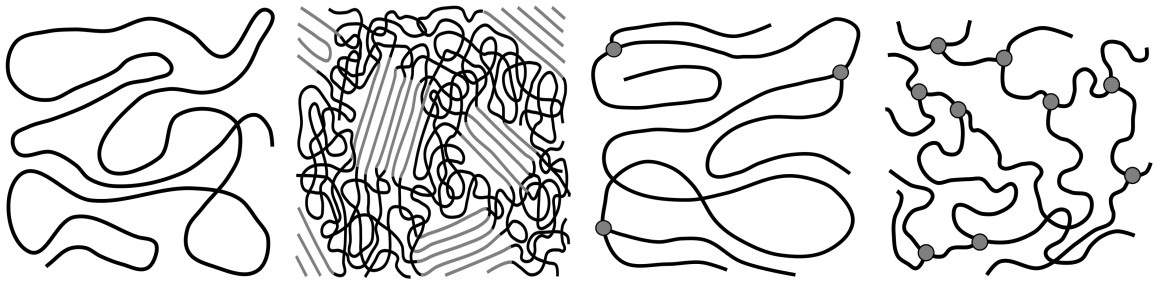


Figure 1.1: Polystyrene composed of styrene monomers (adapted from R.O. Ebewele, Polymer science and technology, CRC press, 2000).

can be chemically bonded, forming cross-linked structures. In contrast, semi-crystalline structures form 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 [8], Z. Tadmor et al., 2006 [9], S. Thomas et al., 2009 [10], O.S. Carneiro et al., 2012 [11], and D.G. Baird et al., 2014 [12]. 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.

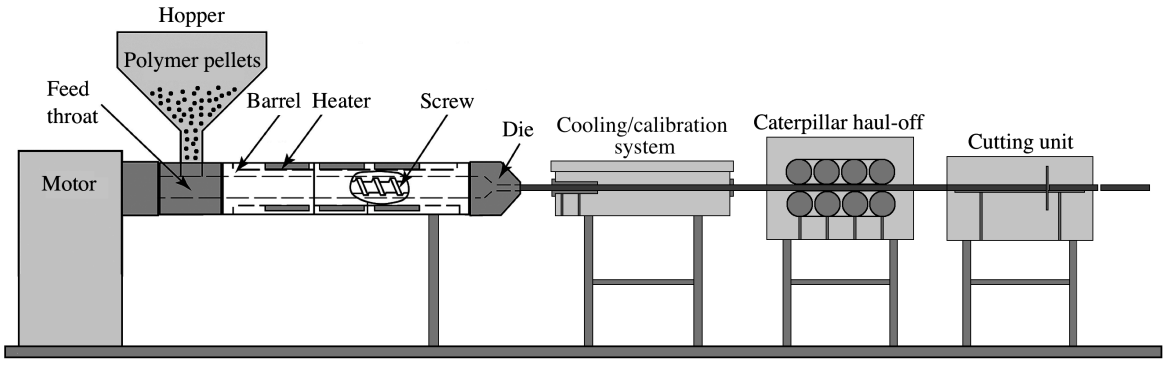


Figure 1.3: Typical thermoplastic profile extrusion line (adapted from C. Rauwendaal, R. Gonzalez-Nunez, D. Rodrigue, Polymer processing: extrusion, in Encyclopedia of polymer science and technology, John Wiley & Sons, 2017).

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.

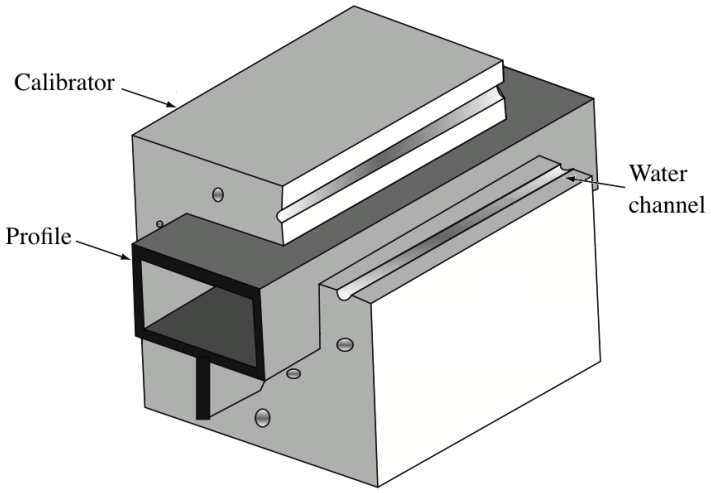


Figure 1.4: Example of a cooling/calibrator system for the thermoplastic profile extrusion (adapted from O.S. Carneiro, J.M. Nóbrega, Design of extrusion forming tools, Smithers Rapra, Shawbury, 2012).

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 [13]. Therefore, an unbalanced flow does not allow to obtain the shape desired for the final 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.

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 [13]. 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,

(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).

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.

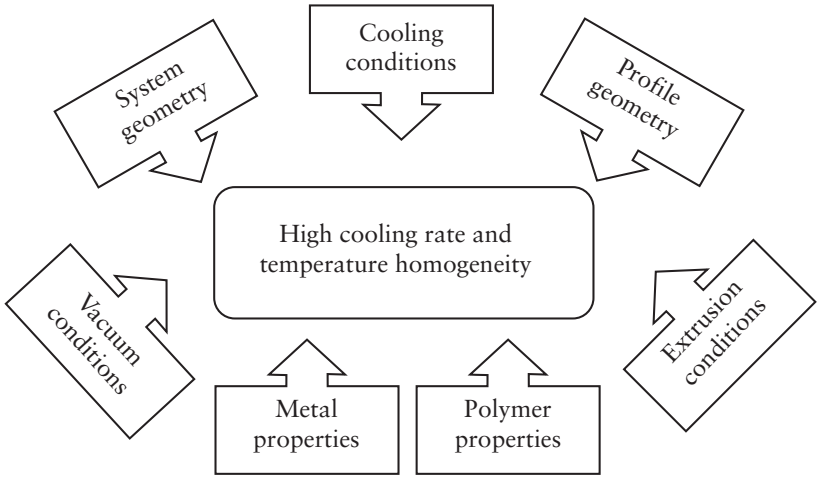


Figure 1.6: Parameters affecting the optimization of the extrusion cooling stage (adapted from O.S. Carneiro, J.M. Nóbrega, Design of extrusion forming tools, Smithers Rapra, Shawbury, 2012).

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 [14–23]. 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 [24]. Additionally, the books of R.R. Huilgol et al., 1997 [25], R.G. Owens et al., 2002 [26], and M.J. Crochet et al., 2012 [27], 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.

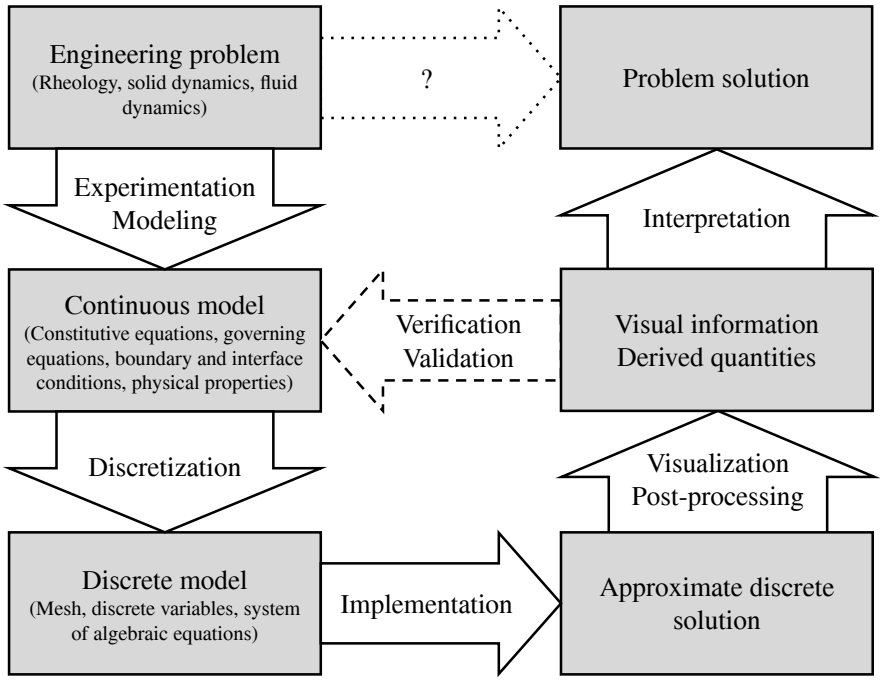


Figure 1.7: The general methodology of the computational modelling for the solution of engineering problems (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

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 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 [76, 77].

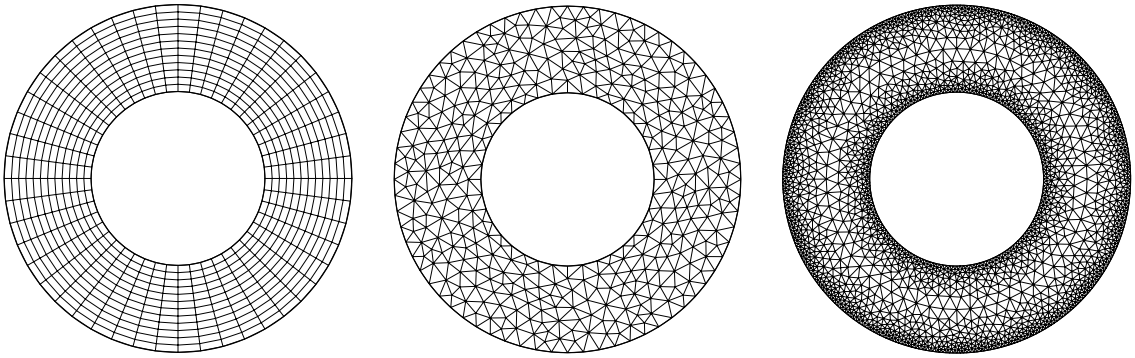
The mathematical models are often complex, comprising systems of partial differential equations intractable analytically when considering geometries with practical interest (otherwise, the computational modelling methodology would be unnecessary). In that regard, proficient numerical methods were developed 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.

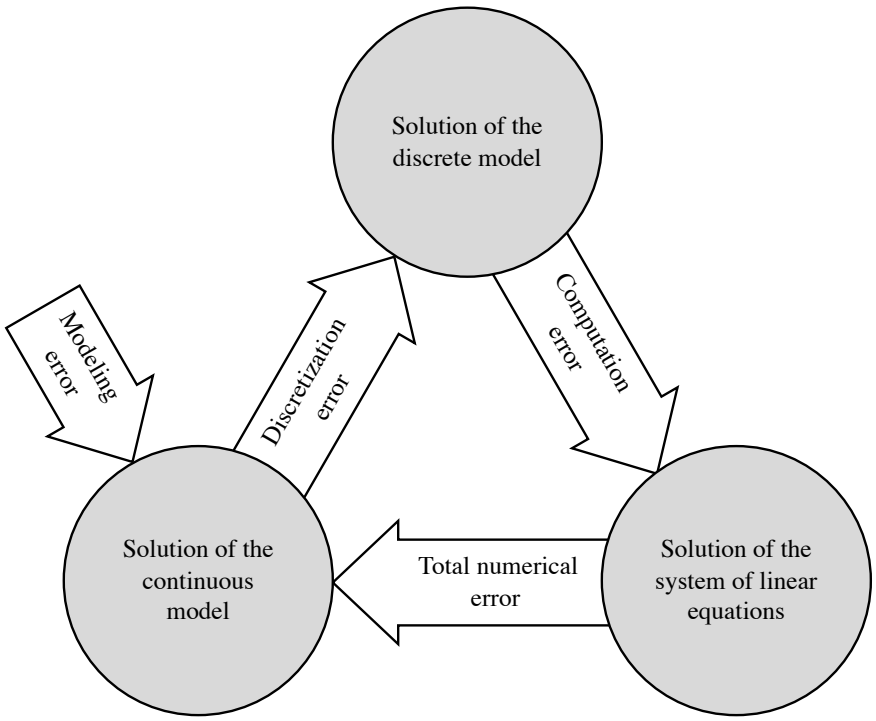


Figure 1.9: The general methodology of the computational modelling for the solution of engineering problems (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

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 [28], C.W. Macosko, 1994 [29], R.G. Larson, 1999 [30], and F.A. Morrison, 2011 [76], 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.

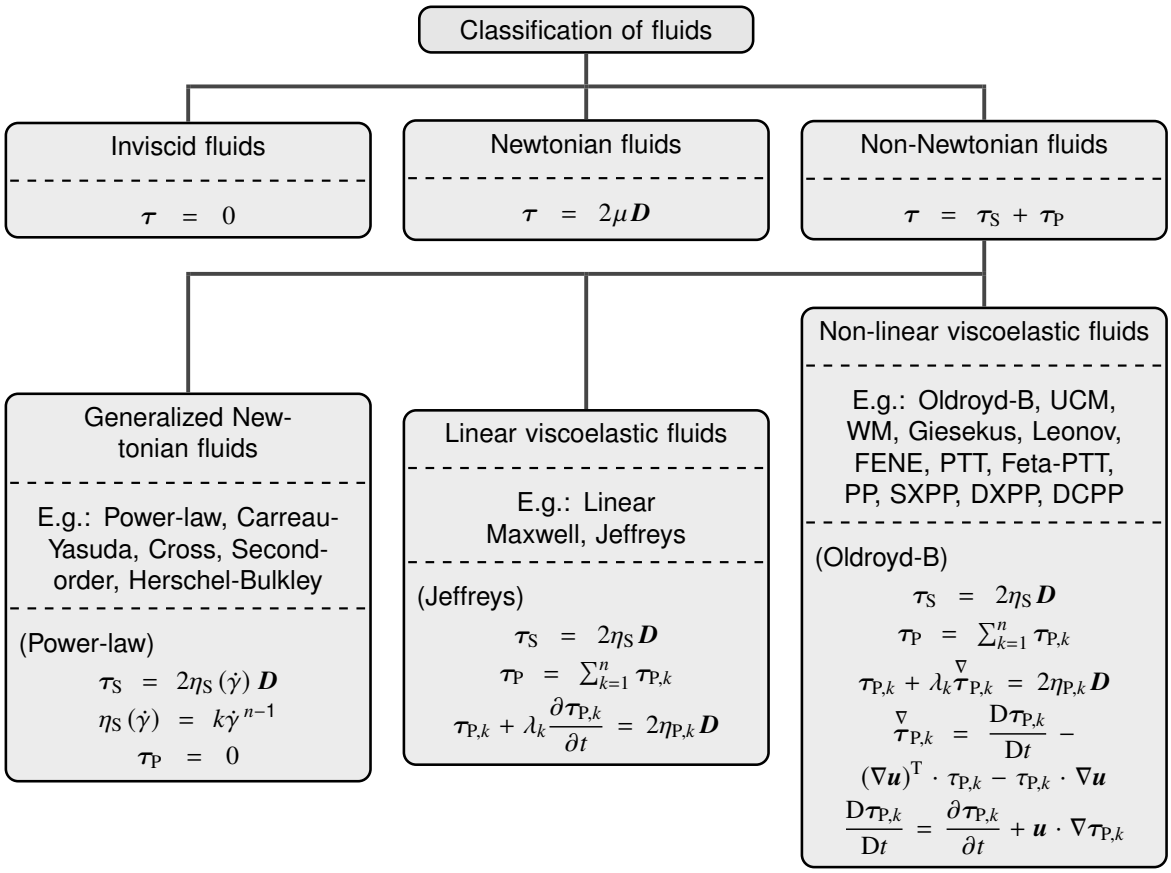


Figure 1.10: Classification and constitutive modelling of fluids.

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 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).

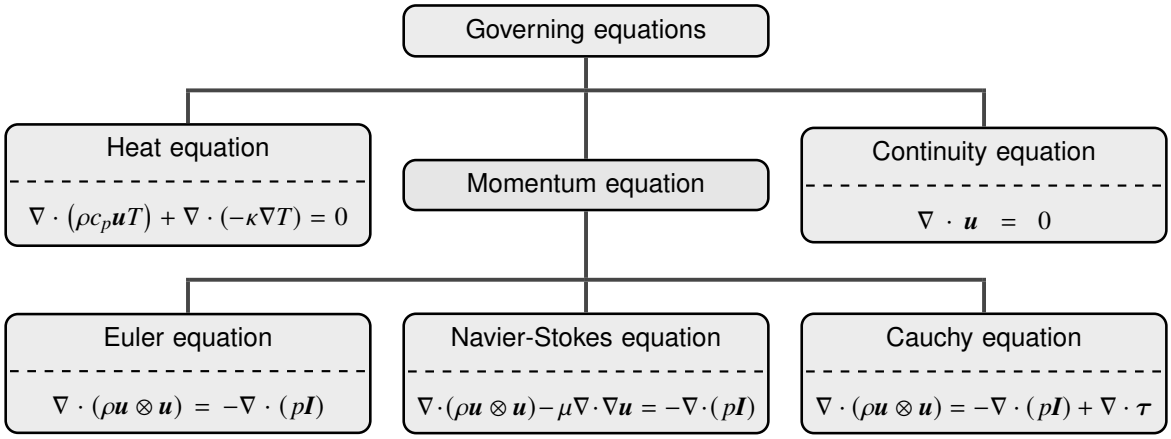


Figure 1.11: Governing equations for steady-state incompressible flows.

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.

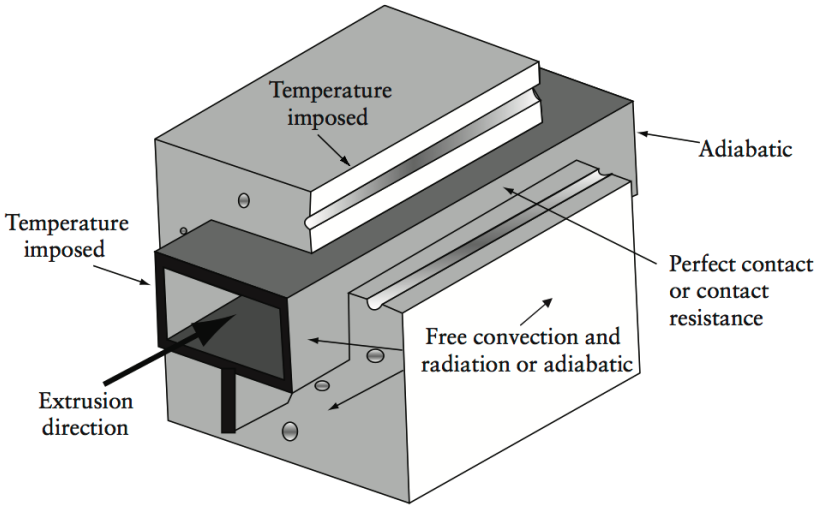


Figure 1.12: Boundary and interface conditions prescribed for the thermoplastic profile extrusion cooling stage with a cooling/calibrator system (adapted from O.S. Carneiro, J.M. Nóbrega, Design of extrusion forming tools, Smithers Rapra, Shawbury, 2012).

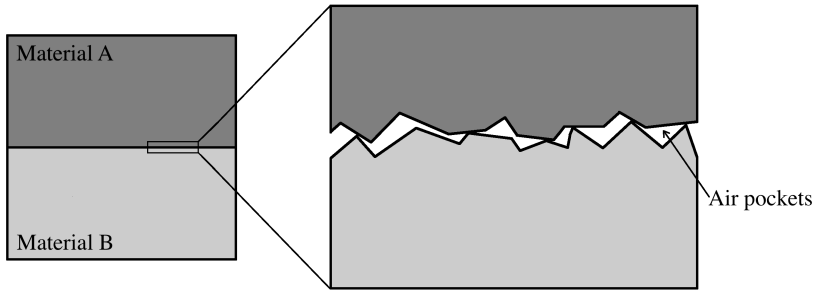


Figure 1.13: Enlargement of the contact at the microscale between two materials.

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

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 [31–33]. 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 [34–38]. 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 [39]. 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 [40], in the American Society of Mechanical Engineers (ASME) and of R.W. MacCormack et al., 1972 [41], and A.W. Rizzi et al., 1973 [42], in the American Institute of Aeronautics and Astronautics (AIAA). Moreover, the seminal works of B.V. Leer, 1973–1979 [43–48], and the works of V.P. Kolgan, 1972–1975 [49–51], 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 [52], and A. Preissmann, 1961 [53], 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 [54], 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 [55], and by A.A. Samarskii, 1965 [56], 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

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

In practice, the finite volume method consists in applying the same procedure considering the

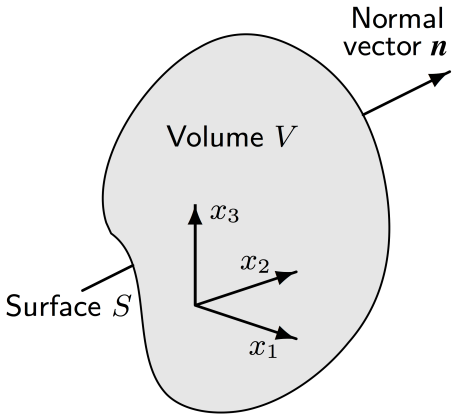
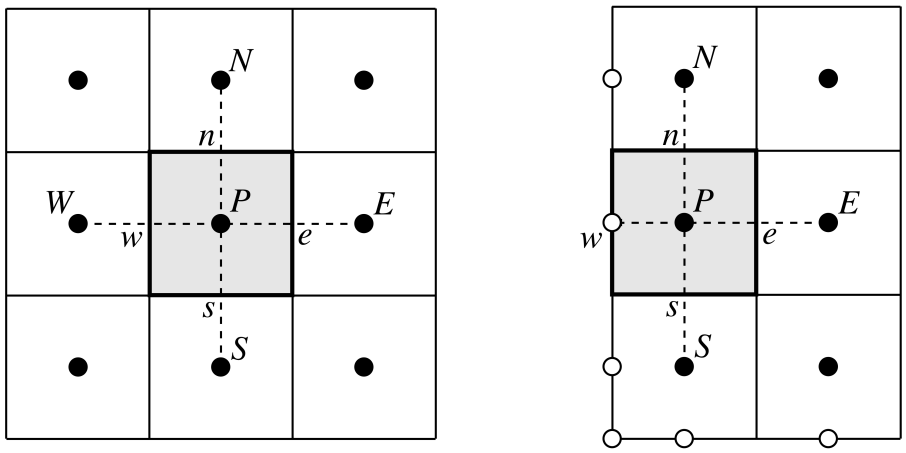


Figure 1.14: Control volume for the finite volume method (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

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

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

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

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

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

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

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

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 [57]. 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}.

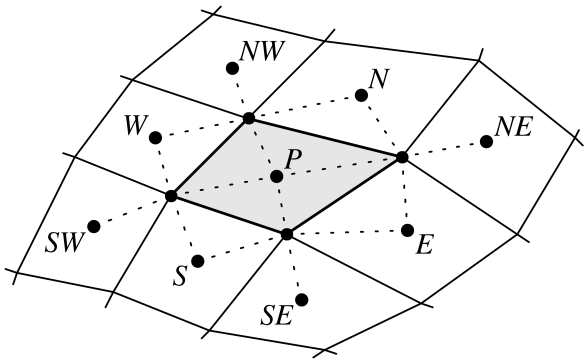


Figure 1.16: Mesh notations for the finite volume method with a structured non-orthogonal mesh (adapted from M. Schäfer, Computational engineering: Introduction to numerical methods, Springer, Berlin, 2006).

Finally, the discrete problem consists in rewritten Equation (1.4) with the numerical

approximations of the surface and volume integrals, given as

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

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 [58–61], 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 [62–64], 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 [65–67], 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 [68, 69] or the mimetic methods [70, 71], among others.

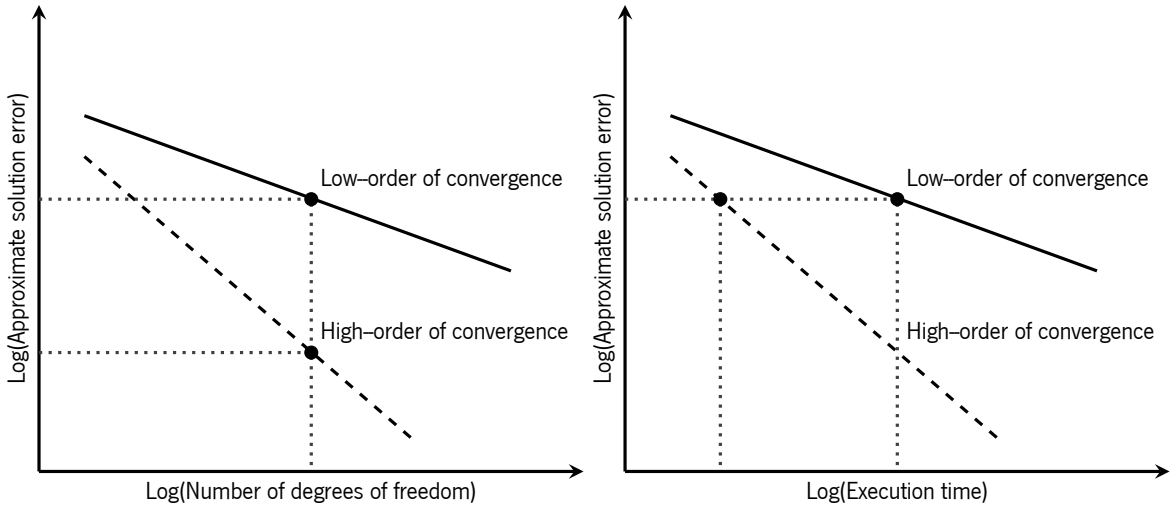
In the case of fluid flow problem, specific numerical techniques are required to correctly solve the div-grad duality between velocity and pressure arising from the balance of momentum and continuity equations [72–75, 78–80]. In that regard, discretization methods are also classified as staggered, when the discrete variables for the velocity and pressure are defined in different meshes (primal and diamond meshes) [81–83], or as collocated, when both discrete variables are defined in the same mesh [84–86]. 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 [87–90].

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.



(a) Number of degrees of freedom. (b) Execution time.

Figure 1.17: Typical convergence curves for low- and very high-orders 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 [91–96]. The method avoids that spurious oscillations are captured due to the Gibbs phenomenon [97], whereas other approaches have been recently developed, as the multi-dimensional optimal order detection method [98–100]. 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.

1.5 Objectives

The computational modelling of ever-increasing complex problems in engineering has only been possible due to the capabilities of modern-day computers and high-performance platforms. Although the hardware development has been crucial to overcome the computational limitations of once intractable engineering problems, the development of proficient numerical methods has significantly contributed to the applicability of the computational modelling. The application of the computational modelling in engineering problems of polymer processing has seen remarkable growth in the past years, providing industries with valuable tools of computer-aided design. Therefore, contributing to the optimization and improvement of the manufacturing technologies and, consequently, also the modernization, competitiveness, and cost reductions of polymer processing industries. However, the complexity of polymer processing applications, generally magnified due to intricate three-dimensional geometries, non-isothermal processes, and polymeric fluids with viscoelastic behaviour, clearly still raise the concern for computational efficiency. Moreover, such a complex scenario that polymer engineers seek to investigate will undoubtedly be more pronounced in the future. Unfortunately, the majority of the computational modelling software commonly used in the polymer processing industry relies on classical numerical methods, which often fail to provide the desired performance and reasonable calculations times in these complex applications. Besides the expensive commercial software licenses, powerful high-performance computer platforms would, in theory, overcome the problem, unfortunately at similarly high costs. Consequently, simplified mathematical models, coarse meshes, or unfinished simulations are the common workarounds, which often succeed to deliver faster results but fail to represent adequately the physical phenomena of interest.

1.5.1 General objectives

The main objective of the present work is to improve the performance, either in terms of numerical accuracy and computational efficiency, of the computational modelling approach of complex problems in polymer processing applications, such as the extrusion. More specifically, the means to achieve the objective is to develop advanced computational methods that allow more accurate, robust, stable, and efficient calculation processes than the classical ones. Moreover, the implementation aspects of the proposed numerical methods are also targeted, developing algorithms to reduce the computational cost of the simulations, also taking advantage of modern multi-core processor computers. Exhaustive and comprehensive benchmarks dedicated to the verification of the proposed computational methods should be carried out, to assess the performance improvement of the computational modelling.

In the present work, the proposed computational methods are applied to the heat transfer and the thermal aspects in polymer processing applications, which already covers a wide range of challenges, such as the optimization of the cooling rate and temperature homogeneity in the extrusion

cooling stage. Nevertheless, the extension of these developments to the computational modelling of the mass transfer and the constitutive modelling, which is also subject of much concern, is straightforward but requires exhaustive verification. Indeed, these mathematical models are mostly based on the same type of partial differential operators, which can be discretized similarly, although usually demanding more elaborated algorithms. In that regard, the present work provides a solid background and a fundamental basis for the future application of the developed computational methods to other mathematical models.

1.5.2 Specific objectives

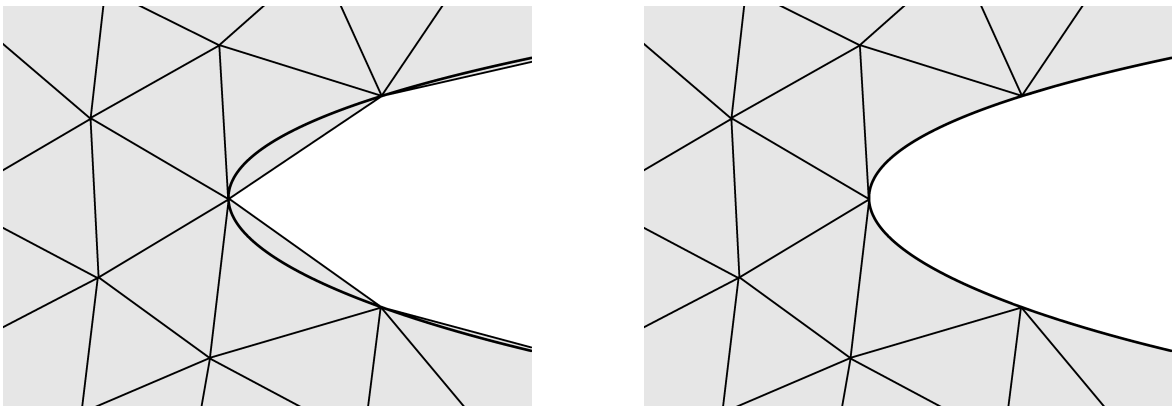
In the present work, a finite volume method with very high-order of convergence will be developed and verified with specific numerical and computational benchmarks to assess its performance. In general terms, the method is based on polynomial reconstructions computed with the least-squares method, which are used to determine approximations to the surface integrals arising in the generic finite volume method formulation. In the case of the heat transfer, these surface integrals correspond to the convective and conductive heat fluxes through the surfaces of the control volumes. The method handles unstructured meshes with general elements, which is a desired characteristic to investigate the complex geometries often arising in polymer processing applications. The method started to gain attention with the works of T.J. Barth et al., 1990 and 1993 [101, 102], although some authors claim that the main ideas and principles are even older. Instead of using finite differences to approximate the surface integrals arising in the generic finite volume formulation, which is severely constrained to the mesh structure, the method is based on polynomial reconstructions. The main idea consists in approximating the discrete variables in the vicinity of a mesh element to a polynomial function, which is computed solving a minimization problem with the least-squares method. Then, the approximation to the surface integrals consists in evaluating the polynomial reconstruction or its derivatives at the points derived from an appropriate quadrature rule. The method has no limitations concerning the mesh structure or the type of mesh elements and requires no orthogonal corrections since the derivatives of the polynomial reconstruction are straightforwardly evaluated. Moreover, the polynomial degree determines the optimal convergence order of the method.

Recent works found in the literature have applied the same ideas in different contexts. For instance, the works of C. Ollivier-Gooch et al., 2002 [103], and of S. Clain et al., 2013 and 2014 [104, 105], investigated the application of the method to the convection-diffusion equation. On the other side, the works of C. Michalak et al., 2009 [106], and of R. Costa et al., 2017 [107, 108], proposed an extension of the method to the Stokes and Navier-Stokes equations. However, these works only concern two-dimensional polygonal domains, whereas the three-dimensional case, with only simple curved geometries, was addressed in the work of A. Boularas et al., 2017 [109], which was later improved in the work of R. Costa et al., 2018 [110]. Besides these essential advances, much research remains to be done towards the application of the method to investigate realistic engineering problems, particularly in polymer processing applications. In that regard, four specific

topics of paramount concern were identified and established as objectives of the present work, as described in the following subsections.

1.5.2.1 General boundary conditions on curved boundaries

The application of very high-order accurate methods to curved domains requires special attention since the geometrical mismatch between the polygonal mesh and the curved boundaries often leads to accuracy and convergence order deterioration. In that regard, the classical technique, either in the finite volume method and finite element method, consists in using curved elements that follow the curved boundary, as illustrated in Figure 1.18. Unfortunately, generating curved meshes requires more elaborated and computationally expensive algorithms, whereas the discretization method often demands complex non-linear transformations and quadrature rules on curved elements. Consequently, the method often becomes cumbersome to apply, in particular in complex three-dimensional geometries. The previous work of R. Costa et al., 2018 [110], proposed a simple and efficient technique that handles arbitrary curved domains with polygonal meshes, therefore overcoming the drawbacks of using curved meshes. However, the work only addresses the Dirichlet boundary condition, which is far from the practical needs of polymer processing applications, as illustrated in Figure 1.12. In that regard, the method will be extended in the present work to general boundary conditions, such as the Neumann and Robin ones, proving a general procedure regardless of the boundary condition type. Further discussion on this topic and literature review is provided in the subsequent chapters.



(a) Polygonal mesh. (b) Curved mesh.

Figure 1.18: Polygonal and curved meshes for a curved boundary.

1.5.2.2 Conjugate heat transfer problems with general interface conditions

Conjugate heat transfer problems often arise in polymer processing applications, namely when two materials are in contact, such as the molten polymeric material and the metallic parts of the machine. These problems require that interface conditions are prescribed on the interface to impose the thermal interaction between the different regions. Moreover, in some polymer processing

applications, imperfect thermal contacts arise due to the microscopic surface roughness of the materials, for instance, in the extrusion, this happens between the molten polymeric material and the cooling/calibrator system, as illustrated in Figure 1.13. The interfacial thermal resistance that arises due to the microscopic air pockets results in temperature jumps, which requires more complex interface conditions than perfect thermal contacts. These issues raise a difficulty, when one wants to preserve the very high-order of convergence, mainly when the problem consists of arbitrary curved interfaces discretized with polygonal meshes, as illustrated in Figure 1.19. In that case, the conservation of conductive heat flux between any two neighbour cells, which is usually intrinsically satisfied in the finite volume method, becomes a concern due to the geometrical mismatch between the polygonal mesh and the curved interface. None of the previous works applies the very high-order accurate method to solve conjugate heat transfer problems with general interface conditions on arbitrary curved interfaces, which is also one of the aims of the present work, considering the developments in the previous topic. Further discussion on this topic and literature review is provided in the subsequent chapters.

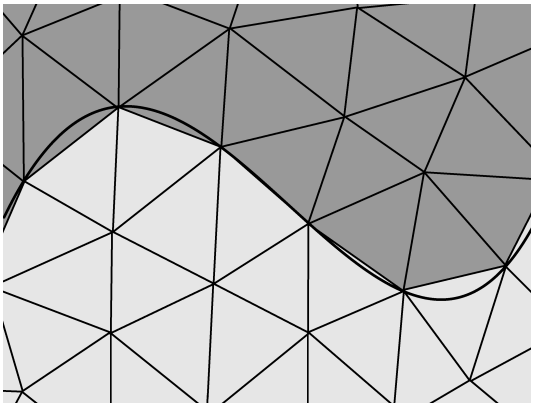


Figure 1.19: Polygonal mesh for a curved interface.

1.5.2.3 Thermal boundary layers in high-Péclet number problems

Boundary layers for the temperature distribution often arise in polymer processing applications as the associated heat transfer problems have typically high Péclet numbers, such as the extrusion. In that case, the convection transport of energy associated with the velocity of the molten polymeric material predominates in the extrusion direction over the conductive transport. The complexity increases in conjugate heat transfer problems, in which the contact between two materials with very distinct physical properties result in thermal boundary layers in the vicinity of the interface. For instance, the molten polymeric material in contact with the cooling/calibrator system, having the former a significantly higher initial temperature and substantially lower thermal conductivity than the latter. Moreover, the thermal boundary layer thickness is proportional to the Péclet number, although it increases as the problem evolves due to the conduction in the opposite direction, as illustrated in Figure 1.20.

The discretization of thermal boundary layer problems usually requires local mesh refinements

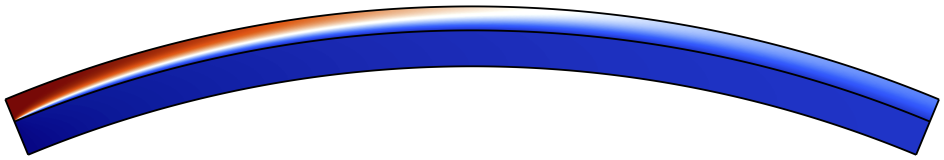


Figure 1.20: Thermal boundary layer on a curved boundary and interface (the colour scale is merely representative).

to capture the high-temperature variations with sufficient accuracy and without spurious oscillations. Moreover, since the temperature variations in thermal boundary layers occur mainly in the perpendicular direction to the boundary or interface, the common practice consists in using stretched meshes, as illustrated in Figure 1.21. Although stretched meshes allow reducing the total number of cells by increasing the mesh size in the tangential direction, very fine meshes are still required in the perpendicular direction where the highest gradients occur. In that context, the very high-order accurate method is capable of maintaining the accuracy of the approximate solution using coarser meshes in both directions, which provides more efficient simulations. However, when associated with arbitrary curved boundaries and interfaces, the problem raises the concern for the capability of the method to preserve the very high-order of convergence with polygonal meshes. On one side, stretched meshes increase the condition number associated to the least-squares matrices of the polynomial reconstructions, consequently impacting on the stability of the method. On the other side, a significant geometrical mismatch arises between the polygonal mesh and the curved boundary or interface, when increasing the aspect ratio of the mesh. Nevertheless, the case of the classical techniques with curved meshes would be even more difficult since inner mesh elements have also to be curved to avoid intersections with the boundary mesh elements. In that regard, the developed very high-order methods should also be verified in the present work with problems having strong thermal boundary layers alongside arbitrary curved boundaries and interfaces, for which stretched meshes with high aspect ratios will be considered. Further discussion on this topic and literature review is provided in the subsequent chapters.

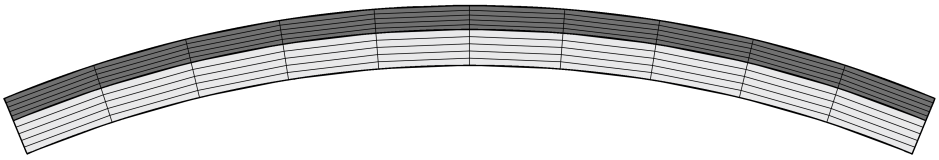


Figure 1.21: Coarse stretched polygonal mesh for a curved boundary and interface.

1.5.2.4 Three-dimensional domains and algorithm optimization

The three-dimensional case is often put aside as it requires very time-consuming simulations, mainly when performing specific numerical benchmarks for the verification of the method. Indeed, the efficiency of the method in the three-dimensional case is critical, whereas the implementation aspects do not often receive much attention and, therefore, computational benchmarks are often avoided.

Moreover, the application of the classical techniques relying on curved meshes to the three-dimensional case with curved boundaries and interfaces becomes cumbersome and often intractable to apply in complex geometries, as illustrated in Figure 1.22. On the other side, the application of the proposed numerical technique with polyhedral meshes is straightforward, implying no additional challenges in comparison to the two-dimensional case. Indeed, polymer processing applications often concern complex three-dimensional geometries and, therefore, these developments are of crucial importance. The work to be performed should also comprise the assessment of three-dimensional case studies considering the developments in the previous topics, aiming at developing optimized algorithms to drastically reduce the computational cost, both in terms of execution time and memory usage, without accuracy deterioration. Further discussion on this topic and literature review is provided in the subsequent chapters.

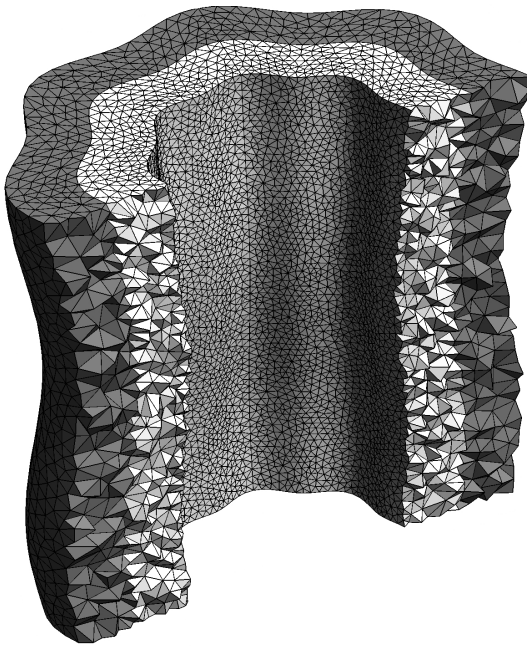


Figure 1.22: Three-dimensional polygonal mesh for a curved boundary.

1.6 Thesis structure

The remaining thesis is divided into five chapters, four of which are built based on already published papers and manuscripts in preparation for publication, which resulted from the work developed.

In Chapter 2, a simple and efficient numerical technique is proposed to allow handling general boundary conditions on arbitrary curved boundaries with a very high-order accurate method. The numerical technique is applied to solve the two-dimensional convection-diffusion equation, whereas a specific numerical benchmark is provided to verify the method with different boundary conditions prescribed on non-trivial curved boundaries.

In Chapter 3, the very high-order accurate method is extended to two-dimensional conjugate heat transfer problems, with general interface conditions on arbitrary curved interfaces. As in the previous chapter, a specific numerical benchmark is provided to verify the method, where interface conditions for both perfect and imperfect thermal contacts are considered on non-trivial curved interfaces.

In Chapter 4, the very high-order accurate method, which is now capable of handling general boundary and interface condition on arbitrary curved domains, is put in place to investigate the cooling stage of the thermoplastic sheet extrusion. The method is firstly assessed with a sanity-check benchmark, whereas a performance comparison between the low- and high-orders of convergence is then carried out in terms of the most relevant parameters for the application.

In Chapter 5, the implementation aspects of the very high-order accurate method are addressed, providing optimized algorithms that substantially reduce the computational cost, both in terms of execution time and memory usage, without accuracy deterioration. The method is exhaustively verified in a numerical benchmark of three-dimensional conjugate heat transfer problems, consisting of curved domains with non-trivial curved boundaries and interfaces. A computational benchmark is also provided to assess the computational gains of the optimized algorithms.

The last chapter of the thesis is dedicated to draw the general conclusions of the work in consideration of the proposed objectives, and also to outline the possibilities of future works.

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 (1997).

[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] D.V. Rosato, Extruding plastics: a practical processing handbook, Springer Science & Business Media, Dordrecht (1998).

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

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

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

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

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

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

[14] 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.

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

[16] 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.

[17] 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.

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

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

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

[25] 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).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[40] 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).

[41] 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).

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

[43] 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).

[44] 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.

[45] 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).

[46] 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.

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

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

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

[55] 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.

[56] 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.

[57] 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.

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

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

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

[66] 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.

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

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

[69] 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.

[70] 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.

[71] 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.

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

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

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

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

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

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

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

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

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

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

[82] 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.

[83] 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.

[84] 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.

[85] 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.

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

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

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

[89] 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.

[90] 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.

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

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

[93] 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.

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

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

[96] 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.

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

[98] 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.

[99] 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.

[100] 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.

[101] T.J. Barth, P.O. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA paper 90–0013 (1990).

[102] T.J. Barth, Recent developments in high order k-exact reconstruction on unstructured meshes, AIAA paper 93–0668 (1993).

[103] 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.

[104] S. Clain, G.J. Machado, J.M. Nóbrega, R.M.S. Pereira, A sixth-order finite volume method for the convection-diffusion problem with discontinuous coefficients, Comput. Meth. Appl. Mech. Eng. 267 (2013) 43–64.

[105] S. Clain, G.J. Machado, A very high-order finite volume method for the time-dependent convection-diffusion problem with Butcher tableau extension, Comput. Math. Appl. 68 (2014) 1292–1311.

[106] C. Michalak, C. Ollivier-Gooch, Unstructured high-order accurate finite-volume solutions of the Navier-Stokes equations, AIAA paper 2009–954 (2009).

[107] R. Costa, S. Clain, G.J. Machado, A sixth-order finite volume scheme for the steady-state incompressible Stokes equations on staggered unstructured meshes, J. Comput. Phys. 349 (2017) 501–527.

[108] R. Costa, S. Clain, G.J. Machado, R. Loulère, A very high-order accurate staggered finite volume scheme for the stationary incompressible Navier-Stokes and Euler equations on unstructured meshes, J. Sci. Comput. 71 (2017) 1375–1411.

[109] A. Boularas, S. Clain, F. Baudoin, A sixth-order finite volume method for diffusion problem with curved boundaries, Appl. Math. Model. 42 (2017) 401–422.

[110] R. Costa, S. Clain, R. Loulère, 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.

CHAPTER 2

Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries

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 [1], which is summarized in the following. The classical approach to handle boundary conditions on curved boundaries is based on the isoparametric element method [2, 3], 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 [4]. 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 [5–7].

R. Costa et al., 2018 [1], 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 [8, 9], 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 $\boldsymbol{n} = \left(n_x, n_y\right) := \left(n_x(\boldsymbol{x}), n_y(\boldsymbol{x})\right)$ stands for the outward unit normal vector to Γ at point $\boldsymbol{x}$ on the boundary (see Figure 2.1).

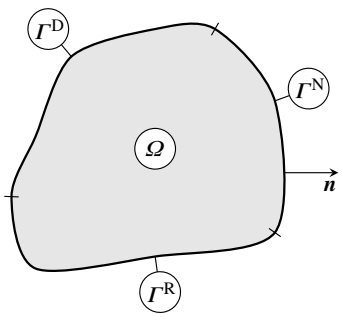


Figure 2.1: Example of arbitrary curved domain with boundary subsets and outward unit normal vector.

2.2.1 Convection-diffusion model

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

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

where $\boldsymbol{u} := (u_x, u_y) := (u_x(\boldsymbol{x}), u_y(\boldsymbol{x}))$ 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 Γ^{D}, Γ^{N}, and Γ^{R} are prescribed with the following boundary conditions:

- On boundary subset Γ^{D}, a Dirichlet boundary condition is prescribed, given as

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

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

- On boundary subset Γ^{N}, a Neumann boundary condition is prescribed, given as

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

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

- On boundary subset Γ^{R}, a Robin boundary condition is prescribed, given as

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

  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 Ω and consists of n non-overlapping convex polygonal cells (triangles, quadrangles, etc.). Cells are denoted as c_i with $i \in \mathcal{I} = \{1, \ldots, n\}$. 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 Γ^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 Γ^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	

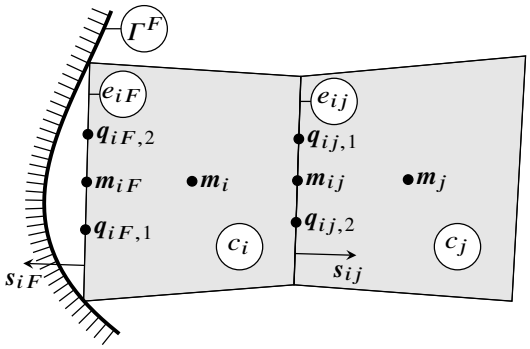


Figure 2.2: Notation and geometric properties for the cells and edges.

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 Ω_Δ, 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, $F \in \{\mathrm{D,N,R}\}$, 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.

2.3 Polynomial reconstructions

The least-squares method for the finite volume method [10–12] is a powerful tool to compute local representations of the underlying solution from scattered pointwise or mean-values associated to the cells. The underlying solution is assumed to be continuous and smooth enough in domain Ω. Hence, no limiting procedure must be designed to avoid parasitical oscillations resulting from the Gibbs phenomenon, such as slope limiting for piecewise linear functions [13–15], (weighted) essentially non-oscillatory ((W)ENO) techniques [16–22], or advanced hierarchical limiting strategies [23, 24] for quadratic or higher degree polynomials. The method consists in computing a local least-squares approximation of function $\phi(\boldsymbol{x})$ targeting some mesh elements, written in the general form as $\varphi(\boldsymbol{x}) = \boldsymbol{\eta}^\mathsf{T} \boldsymbol{p}(\boldsymbol{x})$, where $\boldsymbol{\eta}$ is a vector of coefficients and $\boldsymbol{p}(\boldsymbol{x})$ is a basis function vector. Although any basis function vectors can be considered, polynomial basis functions present a high flexibility and are easy to construct [1, 25, 27]. The local approximations are then used to compute accurate heat flux approximations, and very high-order of convergence can be achieved under mesh refinement.

S. Clain et al., 2013 [25], proposed a polynomial reconstruction method based on a least-squares method to design a very high-order accurate finite volume scheme for convection-diffusion problems.

The scheme uses several types of polynomial reconstructions to compute different kinds of physical heat fluxes. In particular, the authors introduced a conservative polynomial reconstruction associated with the boundary edges to handle the Dirichlet boundary condition. In that regard, the polynomial is rewritten to enforce the prescribed Dirichlet boundary condition at the quadrature points on the boundary edges. As such, the least-squares procedure corresponds to a simple and straightforward unconstrained minimization problem, which can be solved with the normal equations method. In the case of Neumann boundary conditions, no polynomial reconstructions are required as the corresponding boundary heat fluxes at the edge quadrature points are assigned with the provided Neumann boundary condition functions. Despite the effectiveness of the method to provide very high-order of convergence under mesh refinement, the scheme was designed for polygonal domains, which can be fully overlapped by polygonal meshes.

The method introduced by S. Clain et al., 2013 [25], was later extended by R. Costa et al., 2018 [1], for curved domains with prescribed Dirichlet boundary conditions. As in the former, the method adapts the polynomial reconstructions associated to the boundary edges to enforce the boundary conditions at collocation points not belonging to the polygonal edges, successfully restoring the optimal convergence. Since the mesh elements are assumed to be polygonal, Neumann and Robin boundary conditions prescribed on curved boundaries cannot be assigned to the edge quadrature points without accuracy loss. On the other side, the procedure adopted for the Dirichlet boundary edges could be adapted to the cases of Neumann and Robin boundary conditions. That would imply that polynomial representations are rewritten into a different form such that the Neumann or Robin boundary conditions are enforced at the collocation points. Therefore, polynomial $\varphi(\boldsymbol{x})$ is not written simply as $\varphi(\boldsymbol{x}) = \boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{p}_d(\boldsymbol{x})$ but instead in some other way, which in practice turns out to be cumbersome as the transformation yields very complex polynomial reconstruction expressions. Contrarily, this work proposes a new framework for the polynomial reconstructions considering a more sophisticated least-squares procedure. The new approach improves the versatility of the local approximations without loss of accuracy or efficiency, and it can handle genuine general boundary conditions, including the ones of Neumann and Robin types.

The main idea in the polynomial reconstruction method is to collect a stencil of cell mean-values in the vicinity of targeted mesh elements, which are used to compute the coefficients of the polynomial reconstructions in the least-squares sense. Collocation points are defined in the vicinity of the boundary edges to fulfil the boundary conditions, and the associated data (extracted from the prescribed boundary conditions functions) is used to constrain the least-squares procedure and compute coefficients vector $\boldsymbol{\eta}$. Notice that, as the boundary conditions are enforced in the least-squares procedure, polynomial reconstructions are always written as $\varphi(\boldsymbol{x}) = \boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{p}_d(\boldsymbol{x})$ and no cumbersome transformations are needed. In this way, one of the following two types of polynomial reconstructions is used. An unconstrained polynomial reconstruction is applied when only the data in the stencil is used to adjust the polynomial reconstruction. A constrained polynomial reconstruction is applied when prescribed boundary conditions or the cell mean-values conservation need to be exactly fulfilled. Both types of

polynomial reconstructions are obtained employing a weighted, possibly constrained, minimization procedure in the least-squares sense.

2.3.1 Polynomial functions

Polynomial functions are considered to represent the local approximations of the underlying solution. The local polynomial approximation of degree $d > 0$ can be written in a compacted form as

$$\varphi(\boldsymbol{x};\boldsymbol{\eta},\boldsymbol{m}) = \boldsymbol{\eta}^{\mathsf{T}}\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}) = \sum_{\alpha=0}^{d}\sum_{\beta=0}^{d-\alpha}\eta_{(\alpha,\beta)}(x-m_x)^{\alpha}\left(y-m_y\right)^{\beta}, \tag{2.8}$$

where vector $\boldsymbol{p}_d(\boldsymbol{x})$ is a two-dimensional polynomial basis function vector of degree d (see some examples in Table 2.2), and $\boldsymbol{m} = \left(m_x, m_y\right)$ is a reference point used for conditioning purposes. Vector $\boldsymbol{\eta} = \left[\eta_{(\alpha,\beta)}\right]$ gathers the polynomial coefficients associated to the monomial terms. For the sake of simplicity and when it is explicit, the notation $\varphi(\boldsymbol{x}) \coloneqq \varphi(\boldsymbol{x};\boldsymbol{\eta},\boldsymbol{m})$ is adopted to represent the polynomial.

Table 2.2: Monomial terms in the polynomial basis function vector.

	1	x	y	x^2	xy	y^2	x^3	x^2y	xy^2	y^3	x^4	x^3y	x^2y^2	xy^3	y^4
$\boldsymbol{p}_1(\boldsymbol{x})$	✓	✓	✓												
$\boldsymbol{p}_2(\boldsymbol{x})$	✓	✓	✓	✓	✓	✓									
$\boldsymbol{p}_3(\boldsymbol{x})$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
$\boldsymbol{p}_4(\boldsymbol{x})$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

2.3.2 Stencil and weights

To provide an approximation of underlying solution $\phi(\boldsymbol{x})$, a mean-value approximations over each cell c_i, $i \in \mathcal{C}_{\mathcal{M}}$, is considered, given as

$$\phi_i \approx \frac{1}{|c_i|}\int_{c_i}\phi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{2.9}$$

All the N_{C} mean-values are gathered in a vector $\boldsymbol{\Phi} = [\phi_i]_{i\in\mathcal{C}_{\mathcal{M}}}$.

For a given mesh element of reference, a stencil $\mathcal{S}$ is a collection of s cell indices located in its vicinity. A simple algorithm based on cells proximity can be implemented to collect $\mathcal{S}$, but other possibilities can also be considered to handle complex or particular situations. The goal of a stencil is to use the associated data, ϕ_k, $k \in \mathcal{S}$, to compute vector $\boldsymbol{\eta}$ for the local polynomial reconstruction $\varphi(\boldsymbol{x})$. The number of cell indices in the stencil depends on the degree of the polynomial basis function and, eventually, on the type of the polynomial reconstruction to be computed, as seen later. Generally

speaking, a polynomial reconstruction of degree d requires $n = (d+1)(d+2)/2$ coefficients and thus a minimum of n cells has to be collected in stencil $\mathcal{S}$. In accordance to the common practice, a stencil of size $s \approx 1.5n$ is chosen to exceed that minimum, giving some freedom to compute the polynomial reconstruction, which decreases matrices condition numbers and, consequently, improving the procedure robustness [4].

A positive weight, denoted as ω_k, is set for each cell index $k \in \mathcal{S}$ in the stencil. All the weights are gathered in vector $\boldsymbol{\omega} \in \mathbb{R}^s$. The weighting procedure is reported by T.J. Barth, 1992 [26], to compute fluid dynamics solutions with unstructured meshes, with the purpose to reduce the influence of data farther from the local reconstruction. Weights $\omega_k := \omega(d_k; \delta, \gamma)$ are set as the inverse of distance to a power of $\gamma \in \mathbb{R}$, given as

$$\omega(d_k; \delta, \gamma) = \frac{1}{(\delta d_k)^\gamma + 1}, \tag{2.10}$$

where $\delta \in \mathbb{R}_0^+$ is a sensibility factor and $d_k = |\boldsymbol{m}_k - \boldsymbol{m}|$ is the Euclidean distance between cell reference point $\boldsymbol{m}_k$ (the cell centroid) and polynomial reconstruction reference point $\boldsymbol{m}$. The values of δ and γ will be specified later.

2.3.3 Unconstrained polynomial reconstructions

For the unconstrained case, no constraint is applied and the optimal polynomial reconstruction is solely sought in the approximated sense, for which a weighted cost functional from $\mathbb{R}^n$ to $\mathbb{R}$ is considered, given as

$$F(\boldsymbol{\eta}) = \sum_{k \in \mathcal{S}} \omega_k \left[\frac{1}{|c_k|} \int_{c_k} \varphi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \phi_k \right]^2. \tag{2.11}$$

The approximation procedure consists in seeking unique vector $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes functional $F(\boldsymbol{\eta})$ in the least-squares sense, that is

$$\widetilde{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} [F(\boldsymbol{\eta})]. \tag{2.12}$$

An overdetermined system of s linear equations to solve for n unknowns arises, which can be written in the matrix form as $\boldsymbol{WA\eta} = \boldsymbol{Wb}$. Matrix $\boldsymbol{W} \in \mathbb{R}^{s \times s}$ is a diagonal matrix given as $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{\omega})$ and vector $\boldsymbol{b} \in \mathbb{R}^s$ is obtained from the mean-values in $\boldsymbol{\Phi}$ with index in $\mathcal{S}$. Matrix $\boldsymbol{A} \in \mathbb{R}^{s \times n}$ is composed of entries $a_{k,l}$, $k = 1, \ldots, s$, $l = 1, \ldots, n$, given as

$$a_{k,l} = \frac{1}{|c_k|} \int_{c_k} (x - m_x)^\alpha (y - m_y)^\beta \mathrm{d}\boldsymbol{x}, \tag{2.13}$$

where for each index l a pair (α, β), with $\alpha \in \{0, \ldots, d\}$ and $\beta \in \{0, \ldots, d - \alpha\}$, is associated for a

polynomial reconstruction of degree d. Assuming a local indexation of stencil $\mathcal{S} = \{1, 2, \ldots, s\}$ and assuming that vector $\boldsymbol{\eta}$ writes as $\boldsymbol{\eta} = [\eta_1\ \eta_2\ \ldots\ \eta_n]^\mathsf{T}$, the system of linear equations in matrix form then writes as

$$\underbrace{\begin{bmatrix} \omega_1 & 0 & \cdots & 0 \\ 0 & \omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_s \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s,1} & a_{s,2} & \cdots & a_{s,n} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{bmatrix}}_{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} \omega_1 & 0 & \cdots & 0 \\ 0 & \omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_s \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_s \end{bmatrix}}_{\boldsymbol{b}}. \tag{2.14}$$

Finally, the solution of the unconstrained least-squares problem (2.12) provides vector $\widetilde{\boldsymbol{\eta}}$ that minimizes cost functional (2.11) and the associated unconstrained polynomial reconstruction is denoted as $\widetilde{\varphi}(\boldsymbol{x}) = \widetilde{\boldsymbol{\eta}}^\mathsf{T}\boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m})$. Several minimization procedures can be derived to determine the least-squares solution. In this work, the normal equations method is applied.

Following this procedure, unconstrained polynomial reconstruction for each inner edge e_{ij} is computed and denoted as $\varphi_{ij}(\boldsymbol{x}) = \boldsymbol{\eta}_{ij}^\mathsf{T}\boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{ij})$. The reference point corresponds to the edge midpoint, $\boldsymbol{m} := \boldsymbol{m}_{ij}$, and the stencil, $\mathcal{S} := \mathcal{S}_{ij}$, gathers s cells in the vicinity of the edge. The solution of the associated unconstrained least-squares problem (2.12) provides vector $\widetilde{\boldsymbol{\eta}}_{ij}$ that minimizes cost functional (2.11), and the associated unconstrained polynomial reconstruction is denoted as $\widetilde{\varphi}_{ij}(\boldsymbol{x}) = \widetilde{\boldsymbol{\eta}}_{ij}^\mathsf{T}\boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{ij})$.

2.3.4 Constrained polynomial reconstructions

The constrained case arises when it is necessary to exactly fulfil local boundary conditions or preserve the conservation of some physical quantity. In addition to the linear equations to be approximated in the least-squares sense, p linear constraints with respect to vector $\boldsymbol{\eta}$, where $0 < p \leq n$, must be fulfilled. For this purpose, a constraint functional from $\mathbb{R}^n$ to $\mathbb{R}^p$ is introduced and denoted as $G(\boldsymbol{\eta})$, which will be defined later. Mimicking the unconstrained polynomial reconstruction case, the same weighted cost functional from $\mathbb{R}^n$ to $\mathbb{R}$ is considered given as

$$F(\boldsymbol{\eta}) = \sum_{k \in \mathcal{S}} \omega_k \left[\frac{1}{|c_k|} \int_{c_k} \varphi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \phi_k \right]^2. \tag{2.15}$$

The approximation procedure consists in seeking unique vector $\widehat{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes functional $F(\boldsymbol{\eta})$ in the least-squares sense and, at the same time, exactly fulfils $G(\boldsymbol{\eta}) = 0$, that is

$$\widehat{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} [F(\boldsymbol{\eta})] \quad \text{subject to} \quad G(\boldsymbol{\eta}) = 0. \tag{2.16}$$

The least-squares method yields an overdetermined system of s linear equations for n unknowns, which again can be written in the matrix form as $\boldsymbol{WA\eta} = \boldsymbol{Wb}$. Matrices $\boldsymbol{W} \in \mathbb{R}^{s\times s}$ and $\boldsymbol{A} \in \mathbb{R}^{s\times n}$, and vector $\boldsymbol{b} \in \mathbb{R}^{s}$ are given as for the unconstrained polynomial reconstruction (2.14). Moreover, the solution is obtained applying the linear constraints in the matrix form $\boldsymbol{C\eta} = \boldsymbol{d}$ with p equations for n unknowns. Matrix $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ gathers the coefficients of the linear constraints, assumed to be full rank, while vector $\boldsymbol{d} \in \mathbb{R}^{p}$ is the associated right-hand side.

Finally, the solution of the constrained least-squares problem (2.16) provides vector $\widehat{\boldsymbol{\eta}}$ that minimizes cost functional (2.15) and exactly satisfies the linear constraints, and the associated constrained polynomial reconstruction writes as $\widehat{\varphi}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}^{\mathsf{T}}\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m})$. Several minimization procedures fulfilling linear constraints can be found in the literature, and this work considers the so-called linearly constrained Lagrange Multipliers method. This topic is not elaborated, and more details are found in the work of D.P. Bertsekas, 1982 [28].

2.3.4.1 Constrained polynomial reconstructions for cells

Constrained polynomial reconstruction for each cell c_i is computed and denoted as $\varphi_i(\boldsymbol{x}) = \boldsymbol{\eta}_i^{\mathsf{T}}\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_i)$. The unknown polynomial coefficients vector is $\boldsymbol{\eta} := \boldsymbol{\eta}_i$, the reference point, $\boldsymbol{m} = \boldsymbol{m}_i$, is chosen to be the cell centroid, and stencil $\mathcal{S} = \mathcal{S}_i$ gathers s cells in the vicinity of the cell (excluding c_i). The corresponding mean-value, ϕ_i is an approximation of function $\phi(\boldsymbol{x})$ over cell c_i, and to enforce the conservation of ϕ_i in polynomial $\varphi_i(\boldsymbol{x})$, constraint functional $G(\boldsymbol{\eta}) := G_i(\boldsymbol{\eta}_i)$ is given as

$$G_i(\boldsymbol{\eta}_i) = \frac{1}{|c_i|}\int_{c_i}\varphi_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \phi_i. \tag{2.17}$$

Assuming that vector $\boldsymbol{\eta}$ writes as $\boldsymbol{\eta} = [\eta_1\ \eta_2\ \ldots\ \eta_n]^{\mathsf{T}}$, linear constraint $G_i(\boldsymbol{\eta}_i) = 0$, with constraint functional (2.17), writes in matrix form as

$$\underbrace{\begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}}_{\boldsymbol{C}}\underbrace{\begin{bmatrix} \eta_1 & \eta_2 & \ldots & \eta_n \end{bmatrix}^{\mathsf{T}}}_{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} \phi_i \end{bmatrix}}_{\boldsymbol{d}}, \tag{2.18}$$

with coefficients c_l, $l = 1,\ldots,n$, given as

$$c_l = \frac{1}{|c_i|}\int_{c_i}\left(x - m_{i,x}\right)^{\alpha}\left(y - m_{i,y}\right)^{\beta}\mathrm{d}\boldsymbol{x}, \tag{2.19}$$

where for each index l a pair (α,β), with $\alpha \in \{0,\ldots,d\}$ and $\beta \in \{0,\ldots,d-\alpha\}$, is associated for a polynomial reconstruction of degree d.

The solution of the associated constrained least-squares problem (2.16) provides vector $\widehat{\boldsymbol{\eta}}_i$ that minimizes cost functional (2.15) and exactly satisfies linear constraint (2.17), and the associated constrained polynomial reconstruction writes as $\widehat{\varphi}_i(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_i^{\mathsf{T}}\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_i)$.

2.3.4.2 Constrained polynomial reconstructions for boundary edges

Constrained polynomial reconstruction for each boundary edge e_{iF} on computational boundary subset Γ_Δ^F, $F \in \{\mathrm{D},\mathrm{N},\mathrm{R}\}$, with prescribed Dirichlet, Neumann, or Robin boundary conditions, is computed and denoted as $\varphi_{iF}(\boldsymbol{x}) = \boldsymbol{\eta}_{iF}^{\mathsf{T}}\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_{iF})$. The unknown polynomial coefficients vector is $\boldsymbol{\eta} \coloneqq \boldsymbol{\eta}_{iF}$, the reference point, $\boldsymbol{m} \coloneqq \boldsymbol{m}_{iF}$, is chosen to be the edge midpoint, and stencil $\mathcal{S} \coloneqq \mathcal{S}_{iF}$ gathers s cells in the vicinity of the edge. Assume that $\boldsymbol{p}_{iF} = \left(p_{iF,x}, p_{iF,y}\right)$ is a point in the vicinity of the edge with an associated outward unit normal vector $\boldsymbol{v}_{iF} = \left(v_{iF,x}, v_{iF,y}\right)$, to be assigned later.

Constraint functional $G_{iF}(\boldsymbol{\eta}) \coloneqq G\left(\boldsymbol{\eta}_{iF}\right)$ is defined according to the prescribed boundary condition on physical boundary subset Γ^F, as follows:

- For edge $e_{i\mathrm{D}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{D}}$, Dirichlet boundary condition (2.2) is enforced in polynomial $\varphi_{i\mathrm{D}}(\boldsymbol{x})$ with right-hand side defined as $g_{i\mathrm{D}} \coloneqq g^{\mathrm{D}}(\boldsymbol{p}_{i\mathrm{D}})$ and constraint functional $G_{iF}\left(\boldsymbol{\eta}_{iF}\right) \coloneqq G_{i\mathrm{D}}\left(\boldsymbol{\eta}_{i\mathrm{D}}\right)$ given as

$$G_{i\mathrm{D}}\left(\boldsymbol{\eta}_{i\mathrm{D}}\right) = \varphi_{i\mathrm{D}}(\boldsymbol{p}_{i\mathrm{D}}) - g_{i\mathrm{D}}; \tag{2.20}$$

- For edge $e_{i\mathrm{N}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{N}}$, Neumann boundary condition (2.3) is enforced in polynomial $\varphi_{i\mathrm{N}}(\boldsymbol{x})$ with parameter defined as $\kappa_{i\mathrm{N}} \coloneqq \kappa\left(\boldsymbol{p}_{i\mathrm{N}}\right)$, right-hand side defined as $g_{i\mathrm{N}} \coloneqq g^{\mathrm{N}}\left(\boldsymbol{p}_{i\mathrm{N}}\right)$, and constraint functional $G_{iF}\left(\boldsymbol{\eta}_{iF}\right) \coloneqq G_{i\mathrm{N}}\left(\boldsymbol{\eta}_{i\mathrm{N}}\right)$ given as

$$G_{i\mathrm{N}}\left(\boldsymbol{\eta}_{i\mathrm{N}}\right) = -\kappa_{i\mathrm{N}}\nabla\varphi_{i\mathrm{N}}\left(\boldsymbol{p}_{i\mathrm{N}}\right)\cdot\boldsymbol{v}_{i\mathrm{N}} - g_{i\mathrm{N}}. \tag{2.21}$$

- For edge $e_{i\mathrm{R}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{R}}$, Robin boundary condition (2.4) is enforced in polynomial $\varphi_{i\mathrm{R}}(\boldsymbol{x})$ with parameters defined as $\alpha_{i\mathrm{R}} \coloneqq \alpha^{\mathrm{R}}(\boldsymbol{p}_{i\mathrm{R}})$ and $\beta_{i\mathrm{R}} \coloneqq \beta^{\mathrm{R}}(\boldsymbol{p}_{i\mathrm{R}})$, right-hand side defined as $g_{i\mathrm{R}} \coloneqq g^{\mathrm{R}}(\boldsymbol{p}_{i\mathrm{R}})$, and constraint functional $G_{iF}\left(\boldsymbol{\eta}_{iF}\right) \coloneqq G_{i\mathrm{R}}\left(\boldsymbol{\eta}_{i\mathrm{R}}\right)$ given as

$$G_{i\mathrm{R}}\left(\boldsymbol{\eta}_{i\mathrm{R}}\right) = \alpha_{i\mathrm{R}}\varphi_{i\mathrm{R}}(\boldsymbol{p}_{i\mathrm{R}}) + \beta_{i\mathrm{R}}\nabla\varphi_{i\mathrm{R}}(\boldsymbol{p}_{i\mathrm{R}})\cdot\boldsymbol{v}_{i\mathrm{R}} - g_{i\mathrm{R}}. \tag{2.22}$$

Assuming that vector $\boldsymbol{\eta}$ writes as $\boldsymbol{\eta} = [\eta_1\ \eta_2\ \ldots\ \eta_n]^{\mathsf{T}}$, linear constraint $G_{iF}\left(\boldsymbol{\eta}_{iF}\right) = 0$ writes in matrix form as

$$\underbrace{\begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}}_{\boldsymbol{C}} \underbrace{\begin{bmatrix} \eta_1 & \eta_2 & \ldots & \eta_n \end{bmatrix}^{\mathsf{T}}}_{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} g_{iF} \end{bmatrix}}_{\boldsymbol{d}}, \tag{2.23}$$

with coefficients c_l, $l = 1,\ldots,n$, for the Dirichlet, Neumann, and Robin cases given respectively as

$$c_l = \left(p_{i\mathrm{D},x} - m_{i\mathrm{D},x}\right)^{\alpha}\left(p_{i\mathrm{D},y} - m_{i\mathrm{D},y}\right)^{\beta}, \tag{2.24}$$

$$c_l = -\kappa_{i\mathrm{N}} \begin{bmatrix} \alpha \left(p_{i\mathrm{N},x} - m_{i\mathrm{N},x}\right)^{\alpha-1} \left(p_{i\mathrm{N},y} - m_{i\mathrm{N},y}\right)^{\beta} \\ \beta \left(p_{i\mathrm{N},x} - m_{i\mathrm{N},x}\right)^{\alpha} \left(p_{i\mathrm{N},y} - m_{i\mathrm{N},y}\right)^{\beta-1} \end{bmatrix} \cdot \boldsymbol{v}_{i\mathrm{N}}, \tag{2.25}$$

$$\begin{aligned} c_l = {} & \alpha_{i\mathrm{R}} \left(p_{i\mathrm{R},x} - m_{i\mathrm{R},x}\right)^{\alpha} \left(p_{i\mathrm{R},y} - m_{i\mathrm{R},y}\right)^{\beta} \\ & + \beta_{i\mathrm{R}} \begin{bmatrix} \alpha \left(p_{i\mathrm{R},x} - m_{i\mathrm{R},x}\right)^{\alpha-1} \left(p_{i\mathrm{R},y} - m_{i\mathrm{R},y}\right)^{\beta} \\ \beta \left(p_{i\mathrm{R},x} - m_{i\mathrm{R},x}\right)^{\alpha} \left(p_{i\mathrm{R},y} - m_{i\mathrm{R},y}\right)^{\beta-1} \end{bmatrix} \cdot \boldsymbol{v}_{i\mathrm{R}}, \end{aligned} \tag{2.26}$$

where for each index l a pair (α, β), with $\alpha \in \{0, \ldots, d\}$ and $\beta \in \{0, \ldots, d - \alpha\}$, is associated for a polynomial reconstruction of degree d.

The solution of the associated constrained least-squares problem (2.16) provides vector $\widehat{\boldsymbol{\eta}}_{iF}$ that minimizes cost functional (2.15) and exactly satisfies linear constraints (2.20) to (2.22) for a Dirichlet, Neumann, and Robin boundary conditions, respectively, and the associated constrained polynomial reconstruction writes as $\widehat{\varphi}_{iF}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_{iF}^{\mathsf{T}} \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{iF})$.

Table 2.3 summarizes the required polynomial reconstructions associated with the cells, inner edges, and boundary edges and the associated notation and constraint functional.

Table 2.3: Polynomial reconstructions associated with the cells, inner edges, and boundary edges.

Mesh elements	Subset	Notation	Constraint functional
Inner edges	$e_{ij} \subset \Omega_\Delta$	$\widetilde{\varphi}_{ij}(\boldsymbol{x})$	None (unconstrained)
Cells	$c_i \subset \Omega_\Delta$	$\widehat{\varphi}_i(\boldsymbol{x})$	Cell mean-value conservation, Equation (2.17)
Boundary edges	$e_{i\mathrm{D}} \subset \Gamma_\Delta^{\mathrm{D}}$	$\widehat{\varphi}_{i\mathrm{D}}(\boldsymbol{x})$	Dirichlet boundary condition, Equation (2.20)
	$e_{i\mathrm{N}} \subset \Gamma^{\mathrm{N}}$	$\widehat{\varphi}_{i\mathrm{N}}(\boldsymbol{x})$	Neumann boundary condition, Equation (2.21)
	$e_{i\mathrm{R}} \subset \Gamma^{\mathrm{R}}$	$\widehat{\varphi}_{i\mathrm{R}}(\boldsymbol{x})$	Robin boundary condition, Equation (2.22)

2.4 Reconstruction for off-site data

The discretization of regular curved boundaries by polygons yields second-order accurate geometrical approximations [6]. Therefore, if no specific treatment is used to handle the boundary conditions accurately, the scheme usually converges with a maximal second-order of convergence. In this regard, developing techniques dedicated to boundary conditions which are prescribed on curved boundaries is of paramount importance to achieve very high-order of convergence. The classical technique is based on the isoparametric elements method [2, 3], but similar approaches have been designed specifically for the finite volume method. For example, the seminal paper of C. Ollivier-Gooch et al., 2002 [4], introduces a technique to handle smooth curved boundaries based on the constrained least-squares reconstruction method for the boundary edges. The proposed method

enforces the boundary conditions at the quadrature points of the boundaries edges, which are also used to numerically compute the heat fluxes derived from the finite volume formulation [29–31]. In that way, these quadratures and, at the same time, collocation points, have to match the curved physical boundary in order to achieve very high-order of convergence and, therefore, curved cells are required (see Figure 2.3 where $\boldsymbol{q}_{iF,1}$ and $\boldsymbol{q}_{iF,2}$ are quadrature points and $\boldsymbol{n}_{iF}$ is the associated unit normal vectors on the curved cell fitting the curved physical boundary). Moreover, determining the quadrature points and associated outward unit normal vectors for curved edges and cells fitting arbitrary curved boundaries is not a trivial task [27, 29–31], which often demands the use of complex and computational intensive meshing algorithms [6]. Contrary to the isoparametric elements method, the technique proposed in C. Ollivier-Gooch et al., 2002 [4], does not require non-linear transformations. Still, the principal shortcoming remains, namely, the meshing algorithm to generate curved mesh elements fitting curved boundaries and, besides, the high-order accurate quadrature rules for integration on the curved mesh elements. Hence, handling arbitrary two- or three-dimensional curved physical boundaries with such techniques turns out to be a cumbersome task that may result into high computational costs [5–7].

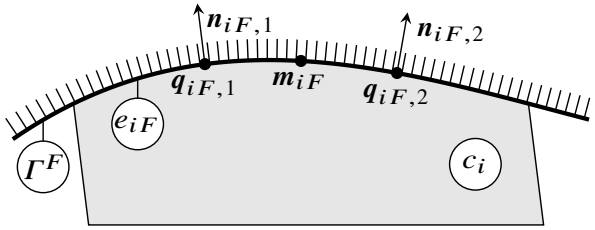
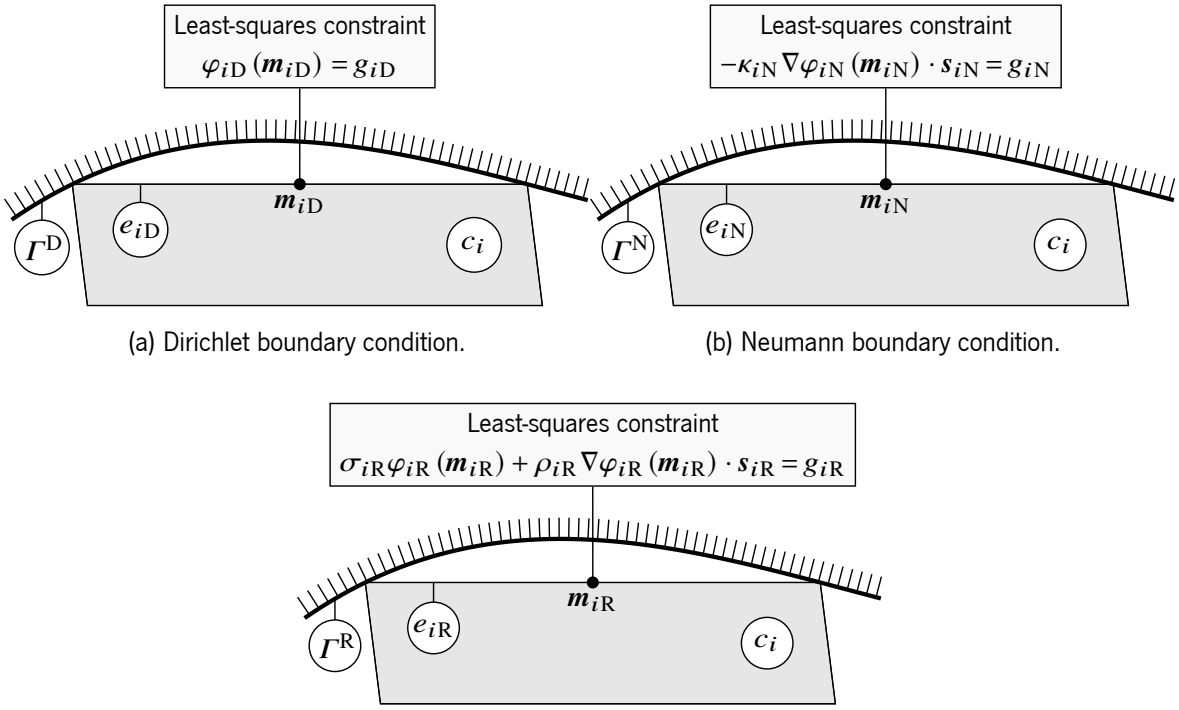


Figure 2.3: Representation of a curved physical boundary fitted with a curved cell.

To avoid the use of curved mesh elements, a new technique proposed by R. Costa et al., 2018 [1], handles solely polygonal representative approximations of curved boundaries. For given boundary edge e_{iF} on computatational boundary subset Γ_{Δ}^{F}, the prescribed boundary condition is evaluated at collocation point $\boldsymbol{p}_{iF}$ with outward unit normal vector $\boldsymbol{v}_{iF}$. Then, a linear constraint is defined at collocation point $\boldsymbol{p}_{iF}$ and is embedded in the least-squares method to compute polynomial reconstruction $\widehat{\varphi}_{iF}(\boldsymbol{x})$ associated to edge e_{iF}. For the Dirichlet, Neumann, and Robin boundary edges, constraint functional $G_{iF}\left(\boldsymbol{\eta}_{iF}\right)$ is defined in Equations (2.20–2.22), respectively. Notice that only one collocation point per boundary edge is required and no exact location for $\boldsymbol{p}_{iF}$ was yet defined but, as stated before, it is sought in the vicinity of the corresponding boundary edge. In the proposed method, $\boldsymbol{q}_{iF,r}$, $r = 1, \ldots, R$, are the quadrature points on straight edge e_{iF} with outward unit normal vector $\boldsymbol{s}_{iF}$. Moreover, the key-point to handle curved boundaries with polygonal mesh elements is to determine the appropriate constraints carefully, such that the polynomial reconstructions fulfil the boundary conditions and provide a very high-order of convergence approximation of the underlying solution. The following sections introduce several of such strategies.

2.4.1 Naive method

A first and straightforward approach consists in constraining the least-squares method for the boundary polynomial reconstructions with a collocation point and associated outward unit normal vector on the computational boundary represented by edge e_{iF} (see Figure 2.4). That is, collocation point $\boldsymbol{p}_{iF} \coloneqq \boldsymbol{m}_{iF}$ (the edge midpoint) and the associated outward unit normal vector $\boldsymbol{v}_{iF} \coloneqq \boldsymbol{s}_{iF}$ (the edge vector) are used in Equations (2.20–2.22). The parameters are defined as $\kappa_{iF} \coloneqq \kappa(\boldsymbol{m}_{iF})$, $\alpha_{iF} \coloneqq \alpha^{\mathrm{R}}(\boldsymbol{m}_{iF})$, and $\beta_{iF} \coloneqq \beta^{\mathrm{R}}(\boldsymbol{m}_{iF})$ and the right-hand side is defined as $g_{iF} \coloneqq g^{F}(\boldsymbol{m}_{iF})$. Such an approach provides a maximal second-order of convergence since collocation point $\boldsymbol{m}_{iF}$ and outward unit normal vector $\boldsymbol{s}_{iF}$ only represent second-order approximations with respect to the curved physical boundary. Therefore, this approach can achieve a very high-order of convergence only if the computational boundaries exactly match the physical ones. In other words, only if the boundaries are polygonal. Notice that collocation point $\boldsymbol{m}_{iF}$ does not belong to the curved physical boundary, hence, an extension of functions $g^{F}(\boldsymbol{x})$, $F \in \{\mathrm{D}, \mathrm{N}, \mathrm{R}\}$, in the vicinity of the edge is required to guarantee that the associated boundary condition makes sense.



(a) Dirichlet boundary condition.

(b) Neumann boundary condition.

(c) Robin boundary condition.

Figure 2.4: Representation of the collocation point and the least-squares constraint for the Naive method.

2.4.2 Reconstruction for off-site data method

The reconstruction for off-site data method (shortened to ROD method) was introduced by R. Costa et al., 2018 [1], to restore the very high-order of convergence for Dirichlet boundary conditions on curved domains. In the present work, the method is extended to general boundary conditions such

as the Neumann and Robin ones. The basic principle is based on an evaluation of the boundary condition with collocation points on the physical boundary. At the same time, the heat flux computation is carried out on the computational boundary. The term off-site data reminds the difference between the data location (boundary condition on physical boundary Γ) and its use (flux computation on computational boundary Γ_Δ). Therefore all computations are performed on the polygonal edges but taking into account the information located on the physical boundary using, for that purpose, the previously introduced constrained polynomial reconstructions for the boundary edges. The main advantages of the ROD method are:

- Numerical integration of the heat fluxes and source terms are only carried out in the polygonal computational domain and not in the curved physical domain.
- No curved mesh element is required.
- No geometrical transformation is required.
- No quadrature point for integration on the curved physical boundary is needed.
- The method is independent of the spatial dimension and the shape of the mesh elements.

To recover the optimal order of convergence and accuracy, the ROD method constrains the least-squares methods with the boundary condition at the curved physical boundary instead of the computational boundary, as in the Naive method (see Figure 2.5 where $\boldsymbol{b}_{iF,1}$ is a collocation point and $\boldsymbol{n}_{iF}$ is the associated unit normal vector on the curved physical boundary). That is, collocation point $\boldsymbol{p}_{iF} := \boldsymbol{b}_{iF}$ matching physical boundary Γ^F and associated outward unit normal vector $\boldsymbol{v}_{iF} := \boldsymbol{n}_{iF} := \boldsymbol{n}(\boldsymbol{b}_{iF})$ to the physical boundary at the same collocation point are determined and used in Equations (2.20–2.22) (see Figure 2.6). The parameters are defined as $\kappa_{iF} := \kappa(\boldsymbol{b}_{iF})$, $\alpha_{iF} := \alpha^{\mathrm{R}}(\boldsymbol{b}_{iF})$, and $\beta_{iF} := \beta^{\mathrm{R}}(\boldsymbol{b}_{iF})$, and the right-hand side is defined as $g_{iF} := g^F(\boldsymbol{b}_{iF})$. In practice, for the sake of simplicity, collocation point $\boldsymbol{b}_{iF}$ can be the orthogonal projection of edge midpoint $\boldsymbol{m}_{iF}$ onto the associated physical boundary Γ^F. Notice that quadrature points $\boldsymbol{q}_{iF,r}$ do belong to the straight edge of the boundary element where all the heat fluxes are computed. This last point definitively distinguishes the ROD method from the one proposed by C. Ollivier-Gooch et al., 2002 [4], where, in the latter, the authors use the same points (quadrature points on the curved edges) both for the boundary condition collocation and for the heat flux integration.

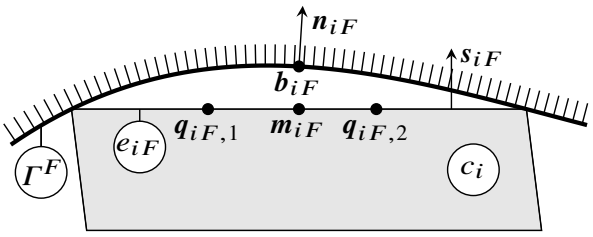
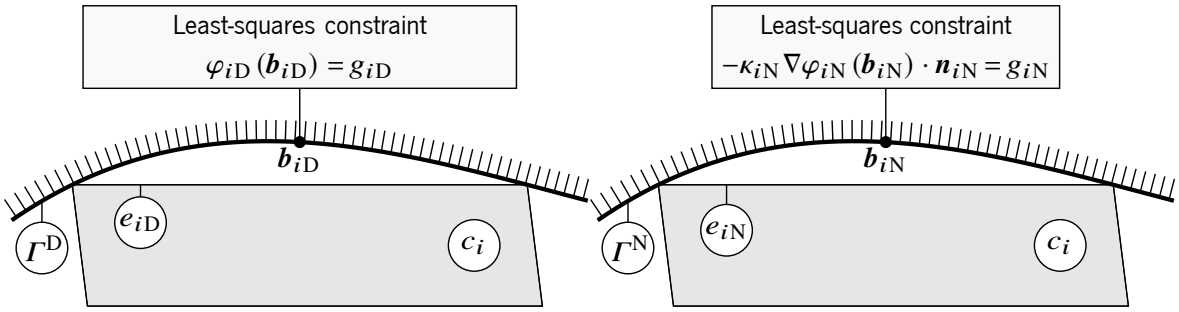
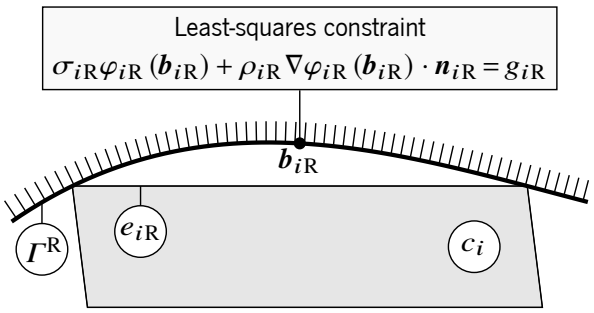


Figure 2.5: Representation of a curved physical boundary subset fitted with a polygonal cell.



(a) Dirichlet boundary condition.

(b) Neumann boundary condition.



(c) Robin boundary condition.

Figure 2.6: Representation of the collocation point and the least-squares constraint for the ROD method.

2.4.3 Adaptive Reconstruction for off-site data method

For the ROD method, the entries of the linear constraints in Equations (2.20–2.22) explicitly depend, by construction, on the physical boundary location via collocation points $\boldsymbol{b}_{iF}$. An alternative method, referred to as adaptive reconstruction for off-site data method (shortened to AROD method), is introduced to handle boundary conditions on curved boundaries. Contrarily to the ROD method, in the AROD method the physical boundary location is decoupled from the least-squares procedure.

Remark 2 One of the advantages of the AROD method over the ROD method lies in the treatment of unsteady and time-dependent problems with moving or tracking boundaries/interfaces problems. In these situations, the displacement of the physical boundary or interface can deteriorate the accuracy of the previously computed polynomial reconstructions associated with the boundary edges and, therefore, limit the convergence orders under mesh refinement. For instance, if the boundary is parameterized by time, $\Gamma(t)$, then collocation points $\boldsymbol{b}_{iF}(t^1)$, used in the ROD method to impose the boundary conditions at time-step $t = t^1$, do not necessarily represent the exact physical boundary at time-step $t = t^2$. Therefore, if the ROD method is applied at time-step $t = t^1$ and if, hypothetically, $\boldsymbol{b}_{iF}(t^2) - \boldsymbol{b}_{iF}(t^1) = \mathcal{O}(h^2)$, then the convergence order at time-step $t = t^2$ is limited to a maximal second-order of convergence. In fact, as the performed least-squares procedure depends on the physical boundary, collocation points $\boldsymbol{b}_{iF}(t^1)$ do not represent exactly the boundary condition at the current time-step, unless this procedure is recomputed for the new collocation points, $\boldsymbol{b}_{iF}(t^2)$. The recomputation of the least-squares procedure is, therefore, required to recover the accuracy and the

convergence order in the ROD method. On the other side, the AROD method can avoid this cost for small perturbations of the physical boundary.

The three main ingredients are:

- A constrained polynomial reconstruction, $\varphi_{iF}(\boldsymbol{x})$, for boundary edge e_{iF} on computatational boundary subset Γ_{Δ}^{F}, with linear constraints (2.20–2.22), depending on the type of the boundary condition (Dirichlet, Neumann, or Robin).

- A collocation point, $\boldsymbol{b}_{iF}$, on physical boundary subset Γ^{F}, where the fitting condition is prescribed, and the associated outward unit normal vector, $\boldsymbol{n}_{iF}$ (for the cases of Neumann and Robin boundary conditions). the parameters in Equations (2.20–2.22) are defined as $\kappa_{iF} \coloneqq \kappa(\boldsymbol{b}_{iF})$, $\alpha_{iF} \coloneqq \alpha^{\mathrm{R}}(\boldsymbol{b}_{iF})$, and $\beta_{iF} \coloneqq \beta^{\mathrm{R}}(\boldsymbol{b}_{iF})$, and the right-hand side is defined as $g_{iF} \coloneqq g^{F}(\boldsymbol{b}_{iF})$.

- A real free-parameter, g, associated to edge midpoint $\boldsymbol{m}_{iF}$, where the least-squares constraint is imposed, and the associated outward unit normal vector, $\boldsymbol{s}_{iF}$ (for the cases of Neumann and Robin boundary conditions). the parameters in Equations (2.20–2.22) are defined as $\kappa_{iF} \coloneqq \kappa(\boldsymbol{m}_{iF})$, $\alpha_{iF} \coloneqq \alpha^{\mathrm{R}}(\boldsymbol{m}_{iF})$, and $\beta_{iF} \coloneqq \beta^{\mathrm{R}}(\boldsymbol{m}_{iF})$, and the right-hand side is defined as $g_{iF} \coloneqq g^{F}(\boldsymbol{m}_{iF})$.

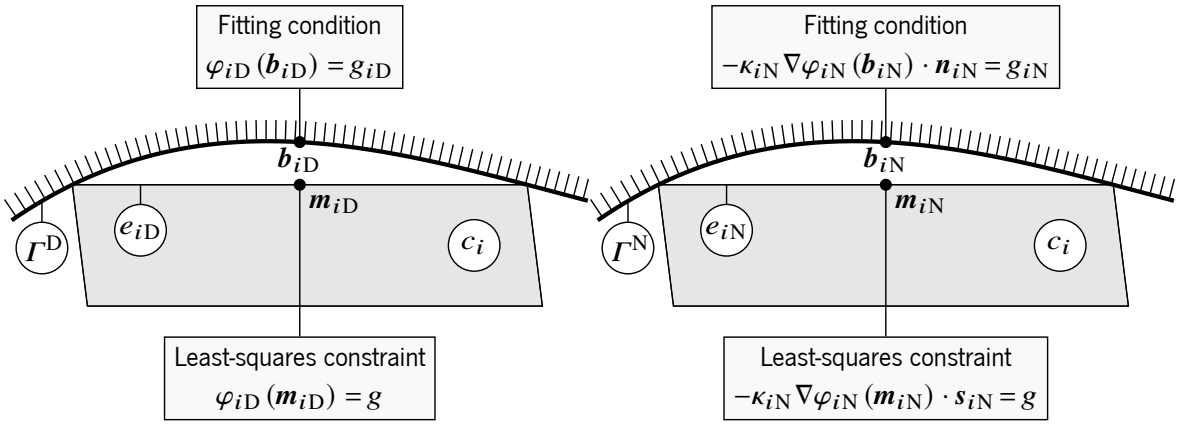
Term fitting condition refers to the condition imposed at collocation point $\boldsymbol{b}_{iF}$ while, on the other side, term least-squares constraint refers to the constraint imposed at edge midpoint $\boldsymbol{m}_{iF}$ (see Figure 2.7). Both conditions are of the same type as the boundary condition type associated to the edge but, while the fitting condition takes the value of the boundary condition function, the least-squares constraint takes the value of free-parameter g. The basic idea of AROD method is to constrain the least-squares method with the least-squares constraint (which does not exactly fulfil the boundary condition at the physical boundary) and then to compute free-parameter g to fulfil the fitting condition.

For boundary edge e_{iF} on computational boundary subset Γ_{Δ}^{F}, polynomial reconstruction $\varphi_{iF}(\boldsymbol{x})$ is computed solving least-squares problem (2.16) with constraint functional $G_{iF}(\boldsymbol{\eta}_{iF})$, which is defined according to the prescribed boundary condition on physical boundary subset Γ^{B}, as follows:

- For edge $e_{i\mathrm{D}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{D}}$, polynomial reconstruction $\varphi_{i\mathrm{D}}(\boldsymbol{x})$ is enforced with $\phi(\boldsymbol{m}_{i\mathrm{D}}) = g$ and constraint functional $G_{iF}(\boldsymbol{\eta}_{iF}) \coloneqq G_{i\mathrm{D}}(\boldsymbol{\eta}_{i\mathrm{D}})$ is given as

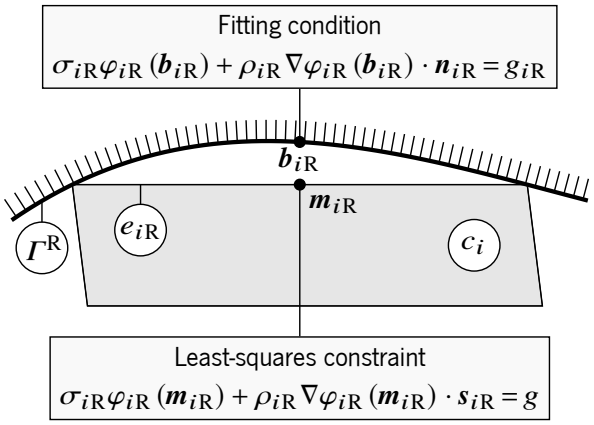
$$G_{i\mathrm{D}}(\boldsymbol{\eta}_{i\mathrm{D}}) = \varphi_{i\mathrm{D}}(\boldsymbol{m}_{i\mathrm{D}}) - g, \tag{2.27}$$

  which corresponds to Equation (2.20) where collocation point $\boldsymbol{p}_{i\mathrm{D}} \coloneqq \boldsymbol{m}_{i\mathrm{D}}$ corresponds to the edge midpoint, and the right-hand side is replaced by free-parameter g (see Figure 2.7(a)).



(a) Dirichlet boundary condition.

(b) Neumann boundary condition.



(c) Robin boundary condition.

Figure 2.7: Representation of the collocation point and the least-squares constraint for the AROD method.

- For edge $e_{i\mathrm{N}}$ on computational boundary subset $\Gamma^{\mathrm{N}}_{\Delta}$, polynomial reconstruction $\varphi_{i\mathrm{N}}(\boldsymbol{x})$ is enforced with $-\kappa(\boldsymbol{m}_{i\mathrm{N}})\nabla\phi(\boldsymbol{m}_{i\mathrm{N}}) \cdot \boldsymbol{s}_{i\mathrm{N}} = g$ and constraint functional $G_{iF}(\boldsymbol{\eta}_{iF}) \coloneqq G_{i\mathrm{N}}(\boldsymbol{\eta}_{i\mathrm{N}})$ is given as

$$G_{i\mathrm{N}}(\boldsymbol{\eta}_{i\mathrm{N}}) = -\kappa_{i\mathrm{N}}\nabla\varphi_{i\mathrm{N}}(\boldsymbol{m}_{i\mathrm{N}}) \cdot \boldsymbol{s}_{i\mathrm{N}} - g, \tag{2.28}$$

  which corresponds to Equation (2.21) where collocation point $\boldsymbol{p}_{i\mathrm{N}} \coloneqq \boldsymbol{m}_{i\mathrm{N}}$ corresponds to the edge midpoint, vector $\boldsymbol{v}_{i\mathrm{N}} \coloneqq \boldsymbol{s}_{i\mathrm{N}}$ corresponds to the edge outward unit normal vector, the parameter is defined as $\kappa_{i\mathrm{N}} \coloneqq \kappa(\boldsymbol{m}_{i\mathrm{N}})$, and the right-hand side is replaced by free-parameter g (see Figure 2.7(b)).

- For edge $e_{i\mathrm{R}}$ on computational boundary subset $\Gamma^{\mathrm{R}}_{\Delta}$, polynomial reconstruction $\varphi_{i\mathrm{R}}(\boldsymbol{x})$ is enforced with $\alpha^{\mathrm{R}}(\boldsymbol{m}_{i\mathrm{R}})\phi(\boldsymbol{m}_{i\mathrm{R}}) + \beta^{\mathrm{R}}(\boldsymbol{m}_{i\mathrm{R}})\nabla\phi(\boldsymbol{m}_{i\mathrm{R}}) \cdot \boldsymbol{s}_{i\mathrm{R}} = g$ and constraint functional $G_{iF}(\boldsymbol{\eta}_{iF}) \coloneqq G_{i\mathrm{R}}(\boldsymbol{\eta}_{i\mathrm{R}})$ is given as

$$G_{i\mathrm{R}}(\boldsymbol{\eta}_{i\mathrm{R}}) = \alpha_{i\mathrm{R}}\varphi_{i\mathrm{R}}(\boldsymbol{m}_{i\mathrm{R}}) + \beta_{i\mathrm{R}}\nabla\varphi_{i\mathrm{R}}(\boldsymbol{m}_{i\mathrm{R}}) \cdot \boldsymbol{s}_{i\mathrm{R}} - g, \tag{2.29}$$

which corresponds to Equation (2.22) where collocation point $\boldsymbol{p}_{i\mathrm{N}} := \boldsymbol{m}_{i\mathrm{N}}$ corresponds to the edge midpoint, vector $\boldsymbol{v}_{i\mathrm{N}} := \boldsymbol{s}_{i\mathrm{N}}$ corresponds to the edge outward unit normal vector, the parameters are defined as $\alpha_{i\mathrm{N}} := \alpha^{\mathrm{R}}(\boldsymbol{m}_{i\mathrm{N}})$ and $\beta_{i\mathrm{N}} := \beta^{\mathrm{R}}(\boldsymbol{m}_{i\mathrm{N}})$, and the right-hand side is replaced by free-parameter g (see Figure 2.7(c)).

Polynomial reconstruction coefficients in vector $\widehat{\boldsymbol{\eta}}_{iF}$ are computed with the constrained least-square procedure and, therefore, linearly depend on free-parameter g. Notice that, the polynomial reconstruction procedure does not depend on the physical boundary but, instead, on the edge midpoint. The fitting condition is introduced as an affine real valued functional $\widehat{\mathcal{B}}_{iF}(g)$ depending on parameter g, and is defined according to the prescribed boundary condition on physical boundary subset Γ^B, as follows:

- For edge $e_{i\mathrm{D}}$ on computational boundary subset $\Gamma^{\mathrm{D}}_{\Delta}$, polynomial reconstruction $\varphi_{i\mathrm{D}}(\boldsymbol{x})$ is enforced with Dirichlet boundary condition $\phi(\boldsymbol{b}_{i\mathrm{D}}) = g^{\mathrm{D}}(\boldsymbol{b}_{i\mathrm{D}})$, and functional $\widehat{\mathcal{B}}_{iF}(g) := \widehat{\mathcal{B}}_{i\mathrm{D}}(g)$ is given as

$$\widehat{\mathcal{B}}_{i\mathrm{D}}(g) = \varphi_{i\mathrm{D}}(\boldsymbol{b}_{i\mathrm{D}}) - g_{i\mathrm{D}}, \tag{2.30}$$

  which corresponds to Equation (2.20) where collocation point $\boldsymbol{p}_{i\mathrm{D}} := \boldsymbol{b}_{i\mathrm{D}}$ matches the physical boundary, and the right-hand side is defined as $g_{i\mathrm{D}} := g^{\mathrm{D}}(\boldsymbol{b}_{i\mathrm{D}})$ (see Figure 2.7(a)).

- For edge $e_{i\mathrm{N}}$ on computational boundary subset $\Gamma^{\mathrm{N}}_{\Delta}$, polynomial reconstruction $\varphi_{i\mathrm{N}}(\boldsymbol{x})$ is enforced with Neumann boundary condition $-\kappa(\boldsymbol{b}_{i\mathrm{N}})\nabla\phi(\boldsymbol{b}_{i\mathrm{N}})\cdot\boldsymbol{n}_{i\mathrm{N}} = g^{\mathrm{N}}(\boldsymbol{b}_{i\mathrm{N}})$ and functional $\widehat{\mathcal{B}}_{iF}(g) := \widehat{\mathcal{B}}_{i\mathrm{N}}(g)$ is given as

$$\widehat{\mathcal{B}}_{i\mathrm{N}}(g) = -\kappa_{i\mathrm{N}}\nabla\varphi_{i\mathrm{N}}(\boldsymbol{b}_{i\mathrm{N}})\cdot\boldsymbol{n}_{i\mathrm{N}} - g_{i\mathrm{N}}, \tag{2.31}$$

  which corresponds to Equation (2.21) where collocation point $\boldsymbol{p}_{i\mathrm{N}} := \boldsymbol{b}_{i\mathrm{N}}$ matches the physical boundary, vector $\boldsymbol{v}_{i\mathrm{N}} := \boldsymbol{n}_{i\mathrm{N}}$ is the associated outward unit normal vector, the parameter is defined as $\kappa_{i\mathrm{N}} := \kappa(\boldsymbol{b}_{i\mathrm{N}})$, and the right-hand side is defined as $g_{i\mathrm{N}} := g^{\mathrm{N}}(\boldsymbol{b}_{i\mathrm{N}})$ (see Figure 2.7(b)).

- For edge $e_{i\mathrm{R}}$ on computational boundary subset $\Gamma^{\mathrm{R}}_{\Delta}$, polynomial reconstruction $\varphi_{i\mathrm{R}}(\boldsymbol{x})$ is enforced with Robin boundary condition $\alpha^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}})\phi(\boldsymbol{b}_{i\mathrm{R}}) + \beta^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}})\nabla\phi(\boldsymbol{b}_{i\mathrm{R}})\cdot\boldsymbol{n}_{i\mathrm{R}} = g^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}})$ and functional $\widehat{\mathcal{B}}_{iF}(g) := \widehat{\mathcal{B}}_{i\mathrm{R}}(g)$ is given as

$$\widehat{\mathcal{B}}_{i\mathrm{R}}(g) = \alpha_{i\mathrm{R}}\varphi_{i\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}}) + \beta_{i\mathrm{R}}\nabla\varphi_{i\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}})\cdot\boldsymbol{n}_{i\mathrm{R}} - g_{i\mathrm{R}}, \tag{2.32}$$

  which corresponds to Equation (2.22) where collocation point $\boldsymbol{p}_{i\mathrm{R}} := \boldsymbol{b}_{i\mathrm{R}}$ matches the physical boundary, vector $\boldsymbol{v}_{i\mathrm{R}} := \boldsymbol{n}_{i\mathrm{R}}$ is the associated outward unit normal vector, the parameters are defined as $\alpha_{i\mathrm{R}} := \alpha^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}})$ and $\beta_{i\mathrm{R}} := \beta^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}})$, and the right-hand side is defined as $g_{i\mathrm{R}} := g^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{R}})$ (see Figure 2.7(c)).

Functional $\widehat{\mathcal{B}}_{iF}(g)$ depends on free-parameter g since polynomial reconstruction $\varphi_{iF}(\boldsymbol{x})$ also depends on it. Notice that the least-squares procedure is constrained with Equations (2.27–2.29) and does not involve directly functional $\widehat{\mathcal{B}}_{iF}(g)$. In order to exactly fulfill the boundary condition, the optimal parameter, denoted as g^*, is sought such that the employed least-squares constraints (2.27–2.29) and the fitting condition, given as $\widehat{\mathcal{B}}_{iF}(g) = 0$, are fulfilled simultaneously. Due to the linearity of functional $\widehat{\mathcal{B}}_{iF}(g)$ with respect to free-parameter g, the optimal parameter is determined as

$$g^* = g_0 - \varepsilon \frac{\widehat{\mathcal{B}}_{iF}(g_0)}{\widehat{\mathcal{B}}_{iF}(g_0 + \varepsilon) - \widehat{\mathcal{B}}_{iF}(g_0)}, \tag{2.33}$$

where g_0 and $\varepsilon \neq 0$ are arbitrary scalars. In other words, the free-parameter in the least-squares constraint is freely adjusted such that $\widehat{\mathcal{B}}_{iF}(g) = 0$ and therefore $\widehat{\varphi}_{iF}(\boldsymbol{b}_{iF}) = g_{iF}$. Although it seems that two constraints are being applied to the least-squares problems, in fact only one constraint is genuinely active, whose right-hand side (the value of g) serves as an intermediate value to fulfil the fitting condition. In practice, scalar $g_0 = g_{iF}$ is chosen for robustness since parameter g^* is relatively close to g_{iF} as only smooth solutions are computed. Moreover, scalar $\varepsilon = 1$ is chosen in this work as only normalized solutions are computed but, for the general case, values of the same order of magnitude as the solution are preferable for robustness purposes.

Contrarily to the previous methods, the polynomial reconstruction $\varphi_{iF}(\boldsymbol{x})$ does not depend on the physical boundary location. Hence, its coefficients are given by a matrix-vector product between the mesh-based structural matrix and the vector of local values ϕ_k, $k \in \mathcal{S}_{iF}$, and depends on free-parameter g. Therefore, the structural matrix is evaluated only during the preprocessing procedure, and the computational effort is reduced. Consequently, the evaluation of functional $\widehat{\mathcal{B}}_{iF}(g)$ is rather fast leading to an efficient computation of g^*. More importantly, AROD method generates a polynomial reconstruction $\widehat{\varphi}_{iF}(\boldsymbol{x})$ which fulfils the prescribed boundary condition at the collocation point on the physical boundary.

Remark 3 The AROD method has been designed for the situation where the physical boundary and its associated collocation points evolve in time. In comparison with the ROD method, the AROD method requires an additional step to compute parameter g^*. For such situations, the ROD method requires to compute the structural matrix again and solve the associated least-squares problem, while the AROD method reuses the initially computed structural matrix.

2.4.4 Adaptive reconstruction for off-site data-XY method

The least-squares constraint and the fitting condition for the AROD method have different collocation points, outward unit normal vectors, and parameters, nonetheless, both are of the same type. The fitting conditions have to agree with the type of the prescribed boundary condition, Dirichlet, Neumann, or Robin. Additionally, the collocation points and outward unit normal vectors are determined from the curved physical boundary to ensure a very high-order of convergence. Remind

that, there is no reason for least-squares constraints (2.27–2.29) to correspond to the prescribed boundary condition. In that regard, a different approach, referred to as adaptive reconstruction for off-site data-XY method (shortened to AROD-XY method), is introduced, where X refers to the fitting condition type and Y refers to the least-squares constraint type. The Dirichlet, Neumann, and Robin types are identified with D, N, or R, respectively.

The formulation of the AROD-XY method strictly follows the AROD method (see Section 2.4.3), where the only change occurs for the least-squares constraint. Consequently, only this difference is addressed. Table 2.4 summarizes the constraints and fitting conditions used for each of the derivations considered from the AROD-XY method.

Table 2.4: Description of the constraints and fitting conditions used for AROD-XY methods.

Method	Fitting condition		Least-squares constraint	
	Type	Equation	Type	Equation
AROD-DN	Dirichlet	(2.30)	Neumann	$G_{i\mathrm{D}}(\boldsymbol{\eta}_{i\mathrm{D}}) = -\kappa(\boldsymbol{m}_{i\mathrm{D}})\nabla\varphi_{i\mathrm{D}}(\boldsymbol{m}_{i\mathrm{D}})\cdot\boldsymbol{s}_{i\mathrm{D}} - g$
AROD-DR			Robin	$G_{i\mathrm{D}}(\boldsymbol{\eta}_{i\mathrm{D}}) = \alpha^{\mathrm{D}}(\boldsymbol{m}_{i\mathrm{D}})\varphi_{i\mathrm{D}}(\boldsymbol{m}_{i\mathrm{D}}) + \beta^{\mathrm{D}}(\boldsymbol{m}_{i\mathrm{D}})\nabla\varphi_{i\mathrm{D}}(\boldsymbol{m}_{i\mathrm{D}})\cdot\boldsymbol{s}_{i\mathrm{D}} - g$
AROD-ND	Neumann	(2.31)	Dirichlet	$G_{i\mathrm{N}}(\boldsymbol{\eta}_{i\mathrm{N}}) = \varphi_{i\mathrm{N}}(\boldsymbol{m}_{i\mathrm{N}}) - g$
AROD-NR			Robin	$G_{i\mathrm{N}}(\boldsymbol{\eta}_{i\mathrm{N}}) = \alpha^{\mathrm{N}}(\boldsymbol{m}_{i\mathrm{N}})\varphi_{i\mathrm{N}}(\boldsymbol{m}_{i\mathrm{N}}) + \beta^{\mathrm{N}}(\boldsymbol{m}_{i\mathrm{N}})\nabla\varphi_{i\mathrm{N}}(\boldsymbol{m}_{i\mathrm{N}})\cdot\boldsymbol{s}_{i\mathrm{N}} - g$
AROD-RD	Robin	(2.32)	Dirichlet	$G_{i\mathrm{R}}(\boldsymbol{\eta}_{i\mathrm{R}}) = \varphi_{i\mathrm{R}}(\boldsymbol{m}_{i\mathrm{R}}) - g$
AROD-RN			Neumann	$G_{i\mathrm{R}}(\boldsymbol{\eta}_{i\mathrm{R}}) = -\kappa(\boldsymbol{m}_{i\mathrm{R}})\nabla\varphi_{i\mathrm{R}}(\boldsymbol{m}_{i\mathrm{R}})\cdot\boldsymbol{s}_{i\mathrm{R}} - g$

Remark 4 When the fitting condition is of Dirichlet or Neumann types, but the least-squares constraint is of Robin type, as in the AROD-DR and AROD-NR methods, functions $\alpha^D(\boldsymbol{x})$, $\beta^D(\boldsymbol{x})$, $\alpha^N(\boldsymbol{x})$, and $\beta^N(\boldsymbol{x})$ are not prescribed from the model on that boundary subset, and, therefore, some definition is required. In this work, functions $\alpha^R(\boldsymbol{x}) := \boldsymbol{u}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x})$ and $\beta^R(\boldsymbol{x}) := -\kappa(\boldsymbol{x})$ are chosen such that the constrained Robin condition represents a total heat flux condition and has a physical meaning.

Remark 5 This approach can improve robustness to handle boundary conditions on curved boundaries, when compared with the AROD method. On the other side, the AROD-XY method can handle situations where the boundary condition type changes, for instance, from one time-step to another in a time-dependent or unsteady problem, therefore avoiding to recompute the least-squares procedure.

Remark 6 In the proposed methods, polynomial reconstructions for each of the boundary edges are enforced with the boundary conditions on a single collocation point, regardless of the convergence order. In contrast, common practice employs multiple collocation points. Hence, each curved physical boundary segment, delimited by the vertices of the boundary edges, is represented solely by one point, which might not seem to be an accurate representation of the whole curved physical boundary to provide a very high-order of convergence. Given that, it is legitimate to question the eventual impact of a single collocation point per boundary polynomial reconstruction on the accuracy and robustness of the

different strategies proposed in this work to handle curved boundaries with polygonal meshes. The first attempt undertaken to devise the proposed method was formulated with multiple collocation points on the curved boundary per boundary polynomial reconstruction. An exhaustive numerical verification was performed varying the number of collocation points from one up to three – notice that a sixth-order of convergence quadrature rule on curved or polygonal line segments requires a minimum of three points. Regardless of the convergence order, the obtained results show no significant and consistent accuracy or robustness improvements when increasing the number of collocation points. In contrast, a slight accuracy and efficiency deterioration occurred in the majority of the tests, when considering multiple collocation points, in particular for Neumann and Robin boundary conditions. Therefore, a single collocation point seems to be sufficient to achieve the very high-order of convergence. These results may be related to the over-fitting and over-constraining of the boundary polynomial reconstructions, which increase the least-squares matrices condition numbers and, consequently, negatively impact the solution accuracy that is not counterbalanced by the additional information. On the other side, remind that each boundary polynomial reconstruction is computed, not only with a single constraint to enforce the boundary condition but also with the mean-values of the neighbour cells, which are fitted in the least-squares sense. In turn, each mean-value is involved in the computation of several boundary polynomial reconstructions since the associated cell is shared among many stencils. Moreover, it is important to notice that, when higher polynomial degrees are considered, larger stencils of cells are required for the computation. Hence, each boundary polynomial reconstruction intrinsically depends, indeed, on several collocation points via other boundary polynomial reconstructions.

2.5 Finite volume discretization

The generic finite volume discretization derives from applying the divergence theorem to Equation (2.1) over each cell c_i, providing the integral equation

$$\int_{\partial c_i} (\boldsymbol{u}(\boldsymbol{x})\phi(\boldsymbol{x}) - \kappa(\boldsymbol{x})\nabla\phi(\boldsymbol{x})) \cdot \boldsymbol{s}_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{c_i} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{2.34}$$

where ∂c_i stands for the cell boundary and $\boldsymbol{s}_i(\boldsymbol{x})$ the associated outward unit normal vector. An R-points Gaussian quadrature rule with weights ξ_r, $r = 1, \ldots, R$, is considered for the line integration, which provides a residual expression of order $2R$ for Equation (2.34), given as

$$\sum_{j \in \mathcal{N}_i} |e_{ij}| \left[\sum_{r=1}^{R} \zeta_r \left(C_{ij,r} + D_{ij,r} \right) \right] - f_i |c_i| = \mathcal{O}\left(h_i^{2R} \right), \tag{2.35}$$

where $h_i = \max_{j \in \mathcal{N}_i} |e_{ij}|$ and f_i stands for an approximation of order $2R$ of the mean-value of source term function $f(\boldsymbol{x})$ over cell c_i. Notice that, if cell c_i is not triangular, a splitting into sub-triangles sharing the cell centroid as a common vertex is used, and the quadrature rule is then applied in each sub-triangle, as described in A. Ern et al., 2009 [32]. Physical convective and conductive heat fluxes

at quadrature point $\boldsymbol{q}_{ij,r}$, denoted respectively as $C_{ij,r}$ and $D_{ij,r}$, are given as

$$C_{ij,r} = \Big(\boldsymbol{u}\Big(\boldsymbol{q}_{ij,r}\Big)\cdot\boldsymbol{s}_{ij}\Big)\phi\Big(\boldsymbol{q}_{ij,r}\Big), \tag{2.36}$$

$$D_{ij,r} = -\kappa\Big(\boldsymbol{q}_{ij,r}\Big)\nabla\phi\Big(\boldsymbol{q}_{ij,r}\Big)\cdot\boldsymbol{s}_{ij}. \tag{2.37}$$

Notice that heat fluxes are intrinsically conservative on the inner edges, that is $C_{ij,r} + C_{ji,r} = 0$ and $D_{ij,r} + D_{ji,r} = 0$.

2.5.1 Numerical heat fluxes

Given the polynomial reconstructions, the approximations to the physical heat fluxes are then computed on the edges, as follows.

For inner edge e_{ij}, numerical convective and conductive heat fluxes $\mathcal{C}_{ij,r}$ and $\mathcal{D}_{ij,r}$ at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1,\ldots,R$, are given as

$$\mathcal{C}_{ij,r} = -\mathcal{C}_{ji,r} = \Big[\boldsymbol{u}\Big(\boldsymbol{q}_{ij,r}\Big)\cdot\boldsymbol{s}_{ij}\Big]^{+}\widehat{\varphi}_i\Big(\boldsymbol{q}_{ij,r}\Big) + \Big[\boldsymbol{u}\Big(\boldsymbol{q}_{ij,r}\Big)\cdot\boldsymbol{s}_{ij}\Big]^{-}\widehat{\varphi}_j\Big(\boldsymbol{q}_{ij,r}\Big), \tag{2.38}$$

$$\mathcal{D}_{ij,r} = -\mathcal{D}_{ji,r} = -\kappa\Big(\boldsymbol{q}_{ij,r}\Big)\nabla\widetilde{\varphi}_{ij}\Big(\boldsymbol{q}_{ij,r}\Big)\cdot\boldsymbol{s}_{ij}, \tag{2.39}$$

where $[a]^{+} = \max(0,a)$ and $[a]^{-} = \min(0,a)$ for any scalar $a \in \mathbb{R}$. Notice that, for the convective heat fluxes the adjacent cell-based constrained polynomial reconstructions are used while for the conductive heat fluxes the edge-based unconstrained polynomial reconstruction is used.

For boundary edge e_{iF} on computational boundary subset Γ_{Δ}^{F}, numerical convective and conductive heat fluxes $\mathcal{C}_{iF,r}$ and $\mathcal{D}_{iF,r}$ at quadrature points $\boldsymbol{q}_{iF,r}$, $r = 1,\ldots,R$, are given as

$$\mathcal{C}_{iF,r} = \big[\boldsymbol{u}(\boldsymbol{q}_{iF,r})\cdot\boldsymbol{s}_{iF}\big]^{+}\widehat{\varphi}_i(\boldsymbol{q}_{iF,r}) + \big[\boldsymbol{u}(\boldsymbol{q}_{iF,r})\cdot\boldsymbol{s}_{iF}\big]^{-}\widehat{\varphi}_{iF}(\boldsymbol{q}_{iF,r}), \tag{2.40}$$

$$\mathcal{D}_{iF,r} = -\kappa(\boldsymbol{q}_{iF,r})\nabla\widehat{\varphi}_{iF}(\boldsymbol{q}_{iF,r})\cdot\boldsymbol{s}_{iF}. \tag{2.41}$$

Notice that, for the convective heat fluxes the adjacent cell-based and edge-based constrained polynomial reconstructions are used. In contrast, for the conductive heat fluxes, the edge-based constrained polynomial reconstruction is used.

Notice that all the numerical heat fluxes are computed on the polygonal boundary edges of the computational domain without any explicit reference to the physical domain. Similarly, the prescribed boundary condition is taken into account via polynomial reconstructions $\widehat{\varphi}_{iF}$ and, therefore, no explicit reference to the boundary condition does appear in the numerical scheme, which handles only two situations, inner or boundary edges.

The constrained polynomial reconstructions associated to the boundary edges are computed with the Naive (see Section 2.4.1), ROD (see Section 2.4.2), AROD (see Section 2.4.3), or AROD-XY

(see Section 2.4.4) methods.

2.5.2 Residual operators

For any vector $\boldsymbol{\Phi}$ in $\mathbb{R}^{N_C}$ of cell mean-value approximations of function $\phi(\boldsymbol{x})$, the residual operator for each cell c_i, defined as

$$\mathcal{R}_i(\boldsymbol{\Phi}) = \sum_{j \in \mathcal{N}_i} |e_{ij}| \left[\sum_{r=1}^{R} \zeta_r \left(\mathcal{C}_{ij,r} + \mathcal{D}_{ij,r} \right) \right] - f_i |c_i|, \tag{2.42}$$

corresponds to the finite volume scheme (2.35) in residual form. Gathering all the residual operators, a global affine operator $\boldsymbol{\Phi} \rightarrow \mathcal{H}(\boldsymbol{\Phi})$ is introduced. Vector $\boldsymbol{\Phi}^\star \in \mathbb{R}^{N_C}$, solution of the system of linear equations $\mathcal{H}(\boldsymbol{\Phi}) = 0$, provides the cell mean-values approximations of the convection-diffusion problem. A GMRES method, supplemented with a preconditioning matrix, is used to compute an approximation of vector $\boldsymbol{\Phi}^\star$, similarly to the proposed methods in S. Clain et al., 2013 [25], and A. Boularas et al., 2017 [27].

2.6 Numerical benchmark

The implementation of the proposed methods is verified, and their performance in terms of accuracy and convergence orders are assessed, for which manufactured solutions are computed for specific curved domains. The associated source term satisfies Equation (2.1) for given convective and conductive coefficient functions. Notice that only smooth solutions of the steady-state two-dimensional convection-diffusion equation are considered to achieve the optimal convergence order. To complete Equation (2.1), Dirichlet, Neumann, or Robin boundary conditions are prescribed on the corresponding boundary subsets, satisfying the associated manufactured solution. The method of verification consists in assessing the order of the error convergence under mesh refinement, applying the different techniques and methods proposed in this work to compute the approximate solution. Polygonal structured and unstructured triangular meshes are considered.

The system of linear equations is solved in the condensed form with the GMRES method where the number of degrees of freedom is $DOF = n^{\mathrm{A}} + n^{\mathrm{B}}$ corresponding to the approximate cell mean-values of the temperature functions. For each cell c_i in computational domain Ω_Δ, the error of approximate cell mean-value ϕ_i^* is determined having the exact cell mean-value given as

$$\overline{\phi}_i = \frac{1}{|c_i|} \int_{c_i} \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{2.43}$$

Then, the relative errors in the L^1- and L^∞-norm, denoted as E_1 and E_∞, respectively, are determined

as

$$E_1 = \frac{\sum_{i=1}^{DOF} \left|\phi_i^* - \overline{\phi}_i\right| |c_i|}{\sum_{i=1}^{DOF} |c_i|}, \tag{2.44}$$

$$E_\infty = \max_{i=1}^{DOF} \left|\phi_i^* - \overline{\phi}_i\right|. \tag{2.45}$$

Consider two meshes for the same domain with different characteristic sizes, a number of degrees of freedom of DOF_1 and DOF_2, respectively, associated relative errors in the L^1-norm of $E_{1,1}$ and $E_{1,2}$, respectively, and associated relative errors in the L^∞-norm of $E_{\infty,1}$ and $E_{\infty,2}$, respectively. Then, the convergence orders for the relative errors in the L^1- and L^∞-norms, denoted as O_1 and O_∞, respectively, are given as

$$O_1 = 2\left|\frac{\ln\left(E_{1,1}/E_{1,2}\right)}{\ln(DOF_1/DOF_2)}\right|, \tag{2.46}$$

$$O_\infty = 2\left|\frac{\ln\left(E_{\infty,1}/E_{\infty,2}\right)}{\ln(DOF_1/DOF_2)}\right|. \tag{2.47}$$

The Naive, ROD, AROD, and AROD-XY methods are tested and compared with polynomial reconstructions of degree d for all the mesh elements, with $d \in \{1,3,5\}$ and tagged as $\mathbb{P}_d$. For the ROD, AROD, and AROD-XY methods, a second possibility will be assessed and consists in considering degree $d+1$ solely for the boundary polynomial reconstructions enforced with a Neumann or Robin boundary conditions. In contrast, degree d is considered for the others, and this case is tagged as $\mathbb{P}_d/\mathbb{P}_{d+1}$.

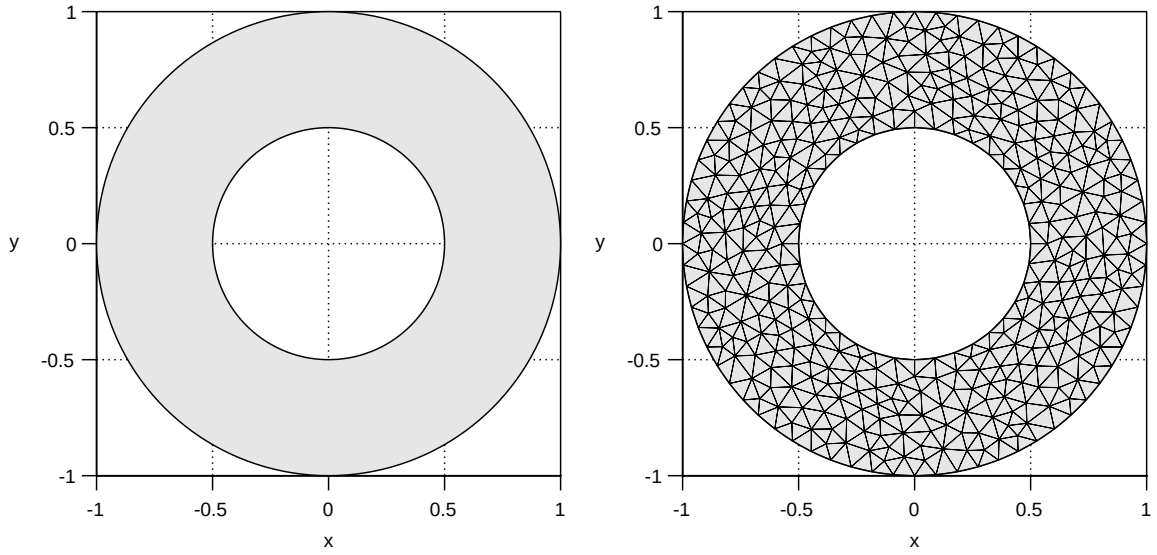
The simulations are carried out considering the weighting function (2.10) with $\delta = 5h$ and $\gamma = 2$, where h is the characteristic size of the reference edge or cell.

Remark 7 For the presented geometries, all the boundary edges intersect the curved physical boundary at only two points, corresponding to the edge vertices. However, this approach has no particular limitation and the boundary edges might intersect at one or several points, or even have no intersection with the physical boundary, provided that the mismatch is of order $\mathcal{O}\left(h^2\right)$ (larger mismatches are not in the scope of this work). The crucial point of the proposed methods is that the collocation points to enforce the boundary conditions have to match the curved physical boundary, regardless of its shape or intersections. Notice, however, that only smooth curved boundaries are handled in this benchmark, and only the two intersection points corresponding to the vertices of the boundary edges are expected for fine enough meshes.

2.6.1 Annular domain test case

A simple annular geometry is first addressed, consisting of an internal, Γ^{I}, and external, Γ^{E}, circumferences centered at point $(0,0)$ with radius $r_{\mathrm{I}} = 0.5$ and $r_{\mathrm{E}} = 1$, respectively (see Figure 2.8).

To perform the simulations, successive finer uniform Delaunay triangular are generated for physical domains Ω (see Figure 2.8). The vertices of the boundary edges coincide with the physical boundaries and all the cells in both meshes are polygonal such that the maximum gap between the physical and the computational boundaries has magnitude order $\mathcal{O}(h^2)$, with h the characteristic mesh size.



(a) Annular domain. (b) Uniform Delaunay triangular mesh.

Figure 2.8: Domain and coarse mesh for the annular domain test case.

The analytic solution for this test case (see Figure 2.9) in polar coordinates (r,θ), with $r^2 = x^2 + y^2$ and $\theta = \arctan(y/x)$, is given as

$$\phi(r,\theta) = a(\exp(R(r)) + \exp(-R(r)) + b) + 1, \quad \text{in } \Omega, \tag{2.48}$$

$$R(r) = \frac{2r - (r_{\mathrm{E}} + r_{\mathrm{I}})}{r_{\mathrm{E}} - r_{\mathrm{I}}}, \quad \text{in } \Omega, \tag{2.49}$$

where $r^2 = x^2 + y^2$ such that $R \in [-1,1]$, and coefficients $a, b \in \mathbb{R}$. Notice that, although the solution is given in terms of r and θ, the problem is numerically solved in Cartesian coordinates. In order to guarantee that $\phi(\boldsymbol{x}) \in [1,2]$ in Ω, coefficients a and b are devised as

$$a = \frac{1}{2 - \exp(1) - \exp(-1)}, \tag{2.50}$$

$$b = -\exp(1) - \exp(-1). \tag{2.51}$$

The associated source term function is obtained after substituting manufactured solution (2.48) into Equation (2.1) (see Figure 2.10).

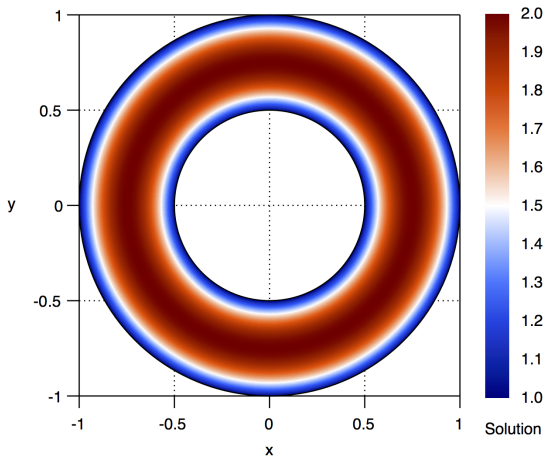
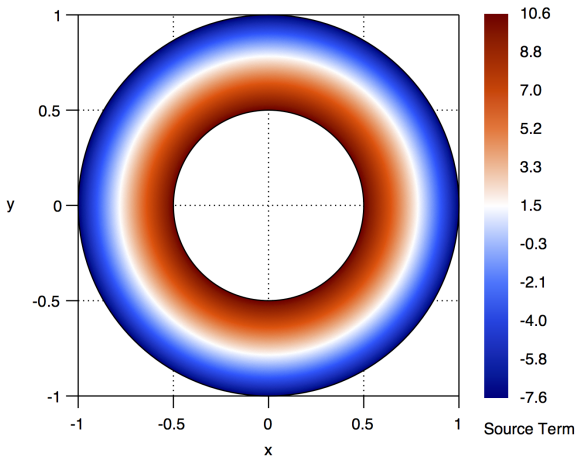
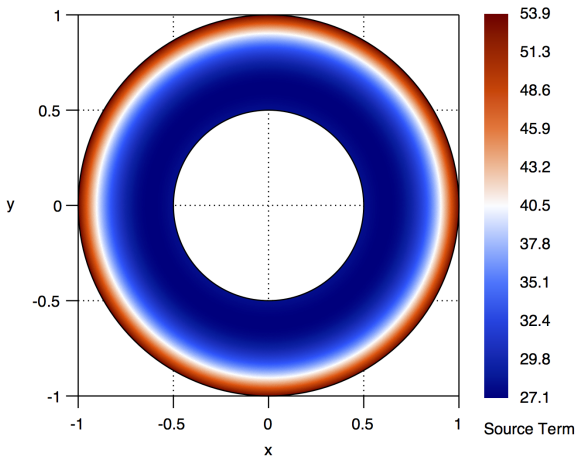


Figure 2.9: Analytic solution for the annular domain test case.



(a) Pure convective case.



(b) Pure conductive case.

Figure 2.10: Source term for the annular domain test case.

Two cases are addressed consisting of a pure convective case with radial velocity function $\boldsymbol{u}(r,\theta) = (\cos(\theta), \sin(\theta))$ and a pure diffusion case with constant thermal conductivity function $\kappa(r,\theta) = 1$. For the former, internal boundary Γ^{I} is prescribed with a constant Dirichlet boundary condition given as $g^{\mathrm{D}}(r,\theta) = 1$, $\boldsymbol{x} \in \Gamma^{\mathrm{I}}$, and therefore $\Gamma^{\mathrm{D}} = \Gamma^{\mathrm{I}}$, whereas no boundary condition is required for external boundary Γ^{E} due to the velocity direction. For the latter, internal boundary Γ^{I} is prescribed with constant Dirichlet boundary condition function $g^{\mathrm{D}}(\boldsymbol{x}) = 1$, while external boundary Γ^{E} is prescribed according to the following cases:

- Dirichlet-Dirichlet boundary conditions: a constant Dirichlet boundary condition with function

$$g^{\mathrm{D}}(r,\theta) = 1, \quad \text{on } \Gamma^{\mathrm{E}}. \tag{2.52}$$

- Dirichlet-Neumann boundary conditions: a constant Neumann boundary condition with function

$$g^{\mathrm{N}}(r,\theta) = 4a(\exp(1)-\exp(-1)), \quad \text{on } \Gamma^{\mathrm{E}}. \tag{2.53}$$

- Dirichlet-Robin boundary conditions: a constant Robin boundary condition with functions

$$g^{\mathrm{R}}(r,\theta) = 1 + 4a(\exp(1)-\exp(-1)), \quad \text{on } \Gamma^{\mathrm{E}}, \tag{2.54}$$

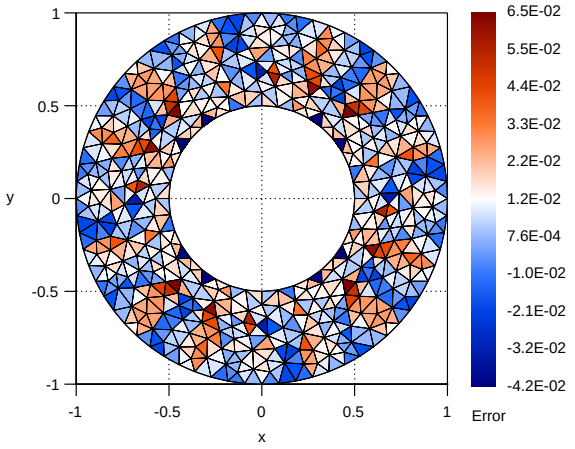
$$\alpha^{\mathrm{R}}(r,\theta) = 1, \quad \text{on } \Gamma^{\mathrm{E}}, \tag{2.55}$$

$$\beta^{\mathrm{R}}(r,\theta) = 1, \quad \text{on } \Gamma^{\mathrm{E}}. \tag{2.56}$$

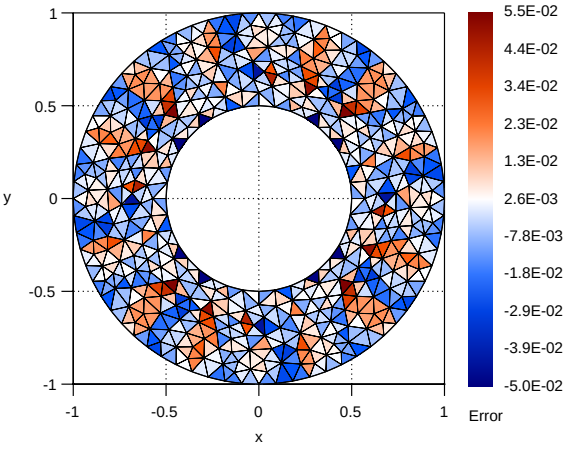
2.6.1.1 Pure convective case

The pure convective case is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1,3,5$, and the Naive and ROD methods are carried out. In Figure 2.11 is shown the errors distribution obtained from the Naive and ROD methods with $\mathbb{P}_d$, $d \in \{1,3,5\}$, polynomial reconstructions, for a uniform Delaunay triangular mesh with 736 cells. As observed, both the Naive and ROD methods seem to provide comparable errors distribution when $\mathbb{P}_1$ polynomial reconstructions are used. In fact, for the Naive method, the geometrical mismatch between the boundary collocation points on the computational boundary and the physical boundary is of order $\mathcal{O}(h^2)$, which is the same order of magnitude as the error for $\mathbb{P}_1$ polynomial reconstructions. On the contrary, when $\mathbb{P}_3$ and $\mathbb{P}_5$ polynomial reconstructions are used, the Naive method noticeably provides a stalled error resulting from the treatment of the internal boundary. Notice that, as there is no diffusion term in this case study and a radial velocity field is prescribed, the external boundary represents an outflow condition and, therefore, no boundary condition and curved boundary treatment is applied here. As predicted, the solution accuracy is restored for high-degree polynomial reconstructions when the ROD method is used since the boundary condition is fulfilled correctly on the internal curved boundary.

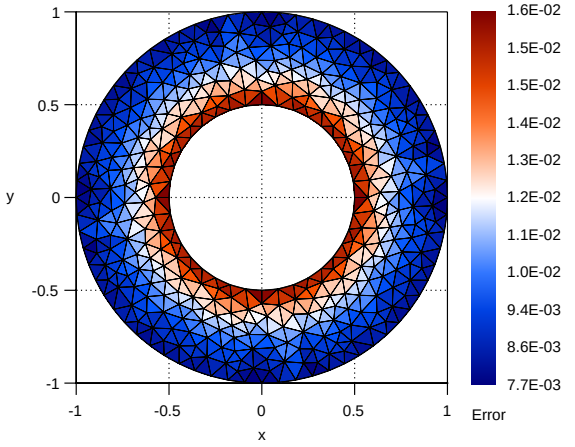
Table 2.5 reports the obtained errors and corresponding convergence orders. As observed, the Naive method provides at most a second-order of convergence order for both error norms, regardless of the polynomial degree considered. Such results are expected attending to the geometrical mismatch of order $\mathcal{O}(h^2)$ between the collocation points on the computational boundary and the physical boundary. Contrarily, the ROD method effectively restores the optimal fourth- and sixth-order of convergence for polynomial degrees $d = 3,5e$, respectively, while no oscillations were obtained. The results support the capability of the ROD method to overcome the second-order limitation expected for those approaches without an appropriate treatment of curved boundaries. Notice that the hyperbolic term is treated without any difficulty, no non-physical oscillations appear due to the upwind heat flux used for the convective contribution, and the steady-state scheme is unconditionally stable with respect to the characteristic mesh size.



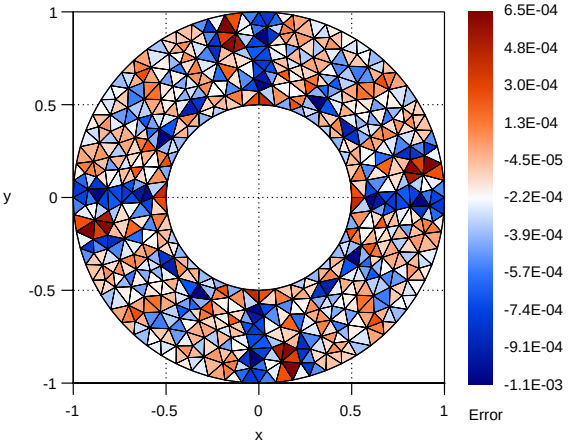
(a) Naive method, polynomial degree $d = 1$.



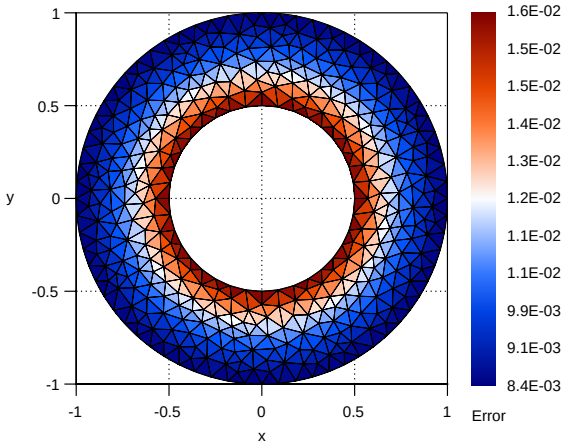
(b) ROD method, polynomial degree $d = 1$.



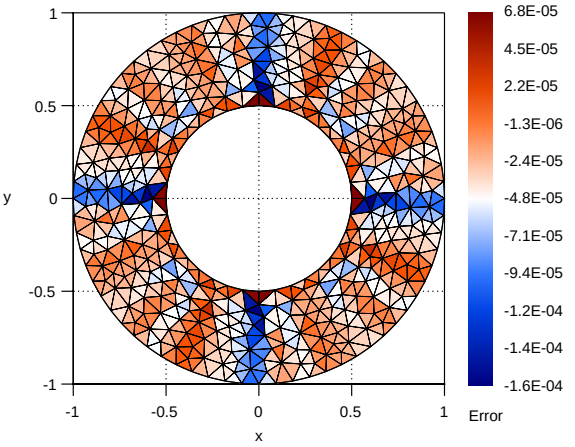
(c) Naive method, polynomial degree $d = 3$.



(d) ROD method, polynomial degree $d = 3$.



(e) Naive method, polynomial degree $d = 5$.



(f) ROD method, polynomial degree $d = 5$.

Figure 2.11: Relative errors distribution obtained from the pure convective case.

Table 2.5: Relative errors and convergence orders obtained in the pure convective case.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive method with $\mathbb{P}_d$												
736	3.65E−03	–	1.48E−02	–	2.72E−03	–	4.11E−03	–	2.78E−03	–	4.05E−03	–
2 828	8.96E−04	2.09	5.25E−03	1.54	7.73E−04	1.87	1.15E−03	1.89	7.79E−04	1.89	1.15E−03	1.87
11 500	2.35E−04	1.91	1.87E−03	1.47	1.94E−04	1.97	2.89E−04	1.97	1.95E−04	1.98	2.90E−04	1.97
45 248	6.37E−05	1.90	6.47E−04	1.55	5.01E−05	1.98	7.49E−05	1.97	5.02E−05	1.98	7.49E−05	1.98
177 880	1.69E−05	1.94	2.21E−04	1.57	1.27E−05	2.00	1.91E−05	2.00	1.27E−05	2.00	1.91E−05	2.00
ROD method with $\mathbb{P}_d$												
736	2.96E−03	–	1.19E−02	–	6.64E−05	–	2.12E−04	–	1.15E−05	–	4.04E−05	–
2 828	7.66E−04	2.01	4.66E−03	1.39	7.19E−06	3.30	2.52E−05	3.17	4.34E−07	4.87	1.25E−06	5.17
11 500	2.00E−04	1.92	1.73E−03	1.42	4.82E−07	3.85	1.95E−06	3.65	7.73E−09	5.74	2.36E−08	5.66
45 248	5.43E−05	1.90	6.09E−04	1.52	3.65E−08	3.77	2.22E−07	3.18	1.71E−10	5.57	6.18E−10	5.32
177 880	1.48E−05	1.90	2.02E−04	1.61	2.38E−09	3.99	1.33E−08	4.11	2.98E−12	5.91	1.41E−11	5.52

2.6.1.2 Pure conductive case

The pure conductive case is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, and the Naive and ROD methods are carried out. The Dirichlet boundary conditions on both boundaries are firstly addressed and Table 2.6 reports the errors and the corresponding convergence orders obtained from the Naive and ROD methods with $\mathbb{P}_d$, $d \in \{1, 3, 5\}$, polynomial reconstructions. Similarly to the pure convective case, the Naive method provides at most a second-order of convergence for both error norms, regardless of the polynomial degree. In contrast, the ROD method restores the optimal convergence orders with no oscillations obtained. Once again, the results highlight the capability of the ROD method to overcome the second-order limitation, in that case for the elliptic term.

The Neumann boundary condition is now prescribed for external boundary Γ^{E} and Table 2.7 reports the errors and the corresponding convergence orders obtained from the Naive method, ROD method with $\mathbb{P}_d$, and $\mathbb{P}_d/\mathbb{P}_{d+1}$, $d \in \{1, 3, 5\}$, polynomial reconstructions. Inexorably the Naive method provides at most a second-order of convergence for both error norms, regardless of the polynomial degree. On the other side, the ROD method with $\mathbb{P}_d$, $d = 1, 3, 5$, polynomial reconstructions achieves second-, third-, and fifth-order of convergence, respectively. In contrast, the second-, fourth-, and sixth-order of convergence were achieved for the previous Dirichlet-Dirichlet boundary conditions case. The scheme is slightly modified to overcome the problem considering $(d+1)$-degree polynomial reconstructions for the Neumann boundary edges while d-degree polynomial reconstructions are kept for the remaining edges and cells. With that approach, a small performance penalty is accepted since additional coefficients for the $\mathbb{P}_{d+1}$ polynomial reconstructions, in comparison with the $\mathbb{P}_d$ ones, have to be computed in the associated least-squares procedure. In practice, this penalty is acceptable

Table 2.6: Relative errors and convergence orders obtained in the pure conductive case with Dirichlet-Dirichlet boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive method with $\mathbb{P}_d$												
736	7.71E−03	—	1.40E−02	—	1.49E−03	—	3.84E−03	—	1.50E−03	—	3.90E−03	—
2 828	2.31E−03	1.79	4.15E−03	1.80	4.06E−04	1.93	1.13E−03	1.82	4.06E−04	1.94	1.13E−03	1.84
11 500	5.90E−04	1.95	1.11E−03	1.89	1.03E−04	1.96	2.87E−04	1.95	1.03E−04	1.96	2.87E−04	1.95
45 248	1.50E−04	2.00	3.09E−04	1.86	2.63E−05	1.99	7.45E−05	1.97	2.63E−05	1.99	7.46E−05	1.97
177 880	3.54E−05	2.11	8.37E−05	1.91	6.65E−06	2.01	1.90E−05	2.00	6.65E−06	2.01	1.90E−05	2.00
ROD method with $\mathbb{P}_d$												
736	7.52E−03	—	1.29E−02	—	3.03E−05	—	1.06E−04	—	3.92E−06	—	1.93E−05	—
2 828	2.21E−03	1.82	3.73E−03	1.84	2.00E−06	4.04	1.24E−05	3.19	1.29E−07	5.07	9.48E−07	4.48
11 500	5.68E−04	1.94	1.01E−03	1.86	2.55E−07	2.94	1.30E−06	3.21	1.16E−09	6.72	9.83E−09	6.51
45 248	1.44E−04	2.00	2.64E−04	1.95	1.84E−08	3.84	8.70E−08	3.95	2.55E−11	5.57	2.74E−10	5.23
177 880	3.38E−05	2.12	7.35E−05	1.87	8.63E−10	4.47	5.53E−09	4.02	5.36E−13	5.64	6.55E−12	5.46

according to the relatively small number of $\mathbb{P}_{d+1}$ polynomial reconstructions, which are around 3–4% of the total number of polynomial reconstructions for uniform Delaunay triangular meshes generated for the annular domain. As seen in Table 2.7, applying the ROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$, $d \in \{1,3,5\}$, polynomial reconstructions, the optimal convergence orders are restored. Notice that the IEEE 745 floating-point precision is attained with polynomial degree $d=5$ for the last mesh and, therefore, it does not make sense to calculate convergence order for this line.

Last, a Robin boundary condition is prescribed on external boundary Γ^{E}, and Table 2.8 reports the errors and corresponding convergence orders obtained from the Naive and ROD methods with $\mathbb{P}_d$ and with $\mathbb{P}_d/\mathbb{P}_{d+1}$, $d \in \{1,3,5\}$, polynomial reconstructions. The results are consistent with the conclusions previously drawn: the Naive method is limited to a second-order of convergence and the ROD method with $\mathbb{P}_d$, $d \in \{3,5\}$, polynomial reconstructions do not achieve the optimal convergence orders. On the other side, the ROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$, $d \in \{3,5\}$, polynomial reconstructions seems to effectively increase the approximated solution accuracy and provides the expected fourth- and sixth-order of convergence, respectively.

2.6.2 Rose-shaped domain test case

The convection-diffusion problem is now addressed in a complex shape the boundaries of which cannot be parameterized by polynomial terms. For this purpose, a diffeomorphism is applied to the annular domain resulting in a rose-shaped domain (see Figure 2.12) as follows. This consists in a

Table 2.7: Relative errors and convergence orders obtained in the pure conductive case with Dirichlet-Neumann boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive method with $\mathbb{P}_d$												
736	5.85E−03	–	1.35E−02	–	2.15E−03	–	3.94E−03	–	2.24E−03	–	4.14E−03	–
2 828	2.16E−03	1.48	4.12E−03	1.76	5.11E−04	2.14	1.12E−03	1.87	5.21E−04	2.16	1.12E−03	1.94
11 500	4.46E−04	2.25	1.09E−03	1.89	1.35E−04	1.90	2.86E−04	1.95	1.37E−04	1.91	2.86E−04	1.95
45 248	1.55E−04	1.54	3.10E−04	1.84	3.42E−05	2.00	7.44E−05	1.96	3.44E−05	2.01	7.44E−05	1.96
177 880	5.24E−05	1.58	1.03E−04	1.60	8.60E−06	2.02	1.90E−05	1.99	8.63E−06	2.02	1.90E−05	1.99
ROD method with $\mathbb{P}_d$												
736	2.27E−03	–	9.39E−03	–	3.27E−05	–	1.15E−04	–	1.93E−05	–	3.65E−05	–
2 828	1.02E−03	1.19	2.96E−03	1.71	1.06E−05	1.67	2.13E−05	2.50	3.86E−07	5.81	8.84E−07	5.53
11 500	1.87E−04	2.42	8.03E−04	1.86	2.39E−06	2.13	4.02E−06	2.38	1.56E−08	4.58	2.86E−08	4.89
45 248	7.99E−05	1.24	2.35E−04	1.80	3.22E−07	2.92	5.33E−07	2.95	5.12E−10	4.99	9.08E−10	5.04
177 880	3.33E−05	1.28	8.41E−05	1.50	3.68E−08	3.17	6.34E−08	3.11	1.64E−11	5.03	3.64E−11	4.70
ROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
736	6.07E−03	–	1.07E−02	–	3.35E−05	–	1.06E−04	–	1.03E−05	–	2.04E−05	–
2 828	2.14E−03	1.54	3.19E−03	1.80	2.44E−06	3.89	1.25E−05	3.18	3.38E−07	5.07	9.78E−07	4.51
11 500	4.56E−04	2.21	9.21E−04	1.77	6.31E−07	1.93	1.33E−06	3.20	3.36E−09	6.57	9.97E−09	6.54
45 248	1.57E−04	1.55	2.97E−04	1.65	5.01E−08	3.70	8.95E−08	3.94	6.93E−11	5.67	2.76E−10	5.24
177 880	5.29E−05	1.59	1.15E−04	1.38	1.39E−09	5.24	5.51E−09	4.07	5.64E−12	–	6.30E−11	–

periodic transformation of its boundaries given in polar coordinates (r,θ) by

$$\Gamma^{\mathrm{I}}:\begin{bmatrix}x\\y\end{bmatrix}=R_{\mathrm{I}}(\theta;r_{\mathrm{I}},\alpha_{\mathrm{I}})\begin{bmatrix}\cos(\theta)\\\sin(\theta)\end{bmatrix}, \tag{2.57}$$

$$\Gamma^{\mathrm{E}}:\begin{bmatrix}x\\y\end{bmatrix}=R_{\mathrm{E}}(\theta;r_{\mathrm{E}},\alpha_{\mathrm{E}})\begin{bmatrix}\cos(\theta)\\\sin(\theta)\end{bmatrix}, \tag{2.58}$$

Functions $R_{\mathrm{I}}(\theta):=R_{\mathrm{I}}(\theta;r_{\mathrm{I}},\alpha_{\mathrm{I}})$ and $R_{\mathrm{E}}(\theta):=R_{\mathrm{E}}(\theta;r_{\mathrm{E}},\alpha_{\mathrm{E}})$ represent the transformed rays of the internal and external boundaries, respectively, having initial constant rays $r_{\mathrm{I}},r_{\mathrm{E}}\in\mathbb{R}$ and coefficients $\alpha_{\mathrm{I}},\alpha_{\mathrm{E}}\in\mathbb{R}$, and are given respectively as

$$R_{\mathrm{I}}(\theta;r_{\mathrm{I}},\alpha_{\mathrm{I}})=r_{\mathrm{I}}\left(1+\frac{1}{20}\sin(\alpha_{\mathrm{I}}\theta)\right), \tag{2.59}$$

$$R_{\mathrm{E}}(\theta;r_{\mathrm{E}},\alpha_{\mathrm{E}})=r_{\mathrm{E}}\left(1+\frac{1}{20}\sin(\alpha_{\mathrm{E}}\theta)\right). \tag{2.60}$$

Table 2.8: Relative errors and convergence orders obtained in the pure conductive case with Dirichlet-Robin boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive method with $\mathbb{P}_d$												
736	6.44E−03	–	1.36E−02	–	1.85E−03	–	3.80E−03	–	1.90E−03	–	3.85E−03	–
2 828	2.16E−03	1.62	4.12E−03	1.78	4.62E−04	2.06	1.12E−03	1.81	4.67E−04	2.08	1.12E−03	1.83
11 500	4.89E−04	2.12	1.10E−03	1.89	1.20E−04	1.92	2.86E−04	1.95	1.21E−04	1.93	2.86E−04	1.95
45 248	1.49E−04	1.74	3.09E−04	1.85	3.05E−05	2.00	7.45E−05	1.96	3.06E−05	2.00	7.45E−05	1.97
177 880	4.44E−05	1.77	9.17E−05	1.78	7.69E−06	2.01	1.90E−05	2.00	7.71E−06	2.01	1.90E−05	2.00
ROD method with $\mathbb{P}_d$												
736	3.96E−03	–	9.97E−03	–	3.00E−05	–	1.10E−04	–	1.08E−05	–	2.24E−05	–
2 828	1.44E−03	1.51	3.03E−03	1.77	6.87E−06	2.19	1.52E−05	2.94	2.19E−07	5.80	9.10E−07	4.76
11 500	3.09E−04	2.19	8.14E−04	1.87	1.51E−06	2.16	2.59E−06	2.52	8.95E−09	4.56	1.75E−08	5.63
45 248	1.02E−04	1.62	2.35E−04	1.81	1.98E−07	2.97	3.25E−07	3.03	2.95E−10	4.98	5.52E−10	5.05
177 880	3.25E−05	1.67	8.43E−05	1.50	2.25E−08	3.18	3.82E−08	3.13	9.84E−12	4.97	4.62E−11	3.62
ROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
736	5.44E−03	–	1.04E−02	–	3.03E−05	–	1.06E−04	–	8.15E−06	–	2.00E−05	–
2828	1.83E−03	1.62	3.08E−03	1.80	1.88E−06	4.13	1.24E−05	3.19	2.58E−07	5.13	9.67E−07	4.50
11500	4.09E−04	2.13	8.42E−04	1.85	4.30E−07	2.11	1.31E−06	3.20	2.54E−09	6.59	9.92E−09	6.53
45248	1.28E−04	1.70	2.47E−04	1.79	3.39E−08	3.71	8.74E−08	3.96	5.26E−11	5.66	2.76E−10	5.23
177880	3.91E−05	1.73	9.53E−05	1.39	5.55E−10	6.01	5.54E−09	4.03	1.73E−11	–	1.11E−10	–

The resulting geometry is designated rose-shaped domain because it resembles rose petals. In this case study, initial rays $r_\mathrm{I} = 0.5$ and $r_\mathrm{E} = 1$ and coefficients $\alpha_\mathrm{I} = 5$ and $\alpha_\mathrm{E} = 8$ are considered. Notice that the annular geometry is recovered assigning $\alpha_\mathrm{I} = \alpha_\mathrm{E} = 0$. Outward unit normal vectors on the internal and external boundaries, $\boldsymbol{n}_\mathrm{I} \coloneqq \boldsymbol{n}_\mathrm{I}(\theta)$ and $\boldsymbol{n}_\mathrm{E} \coloneqq \boldsymbol{n}_\mathrm{E}(\theta)$, respectively, are obtained applying the chain rule and are given as

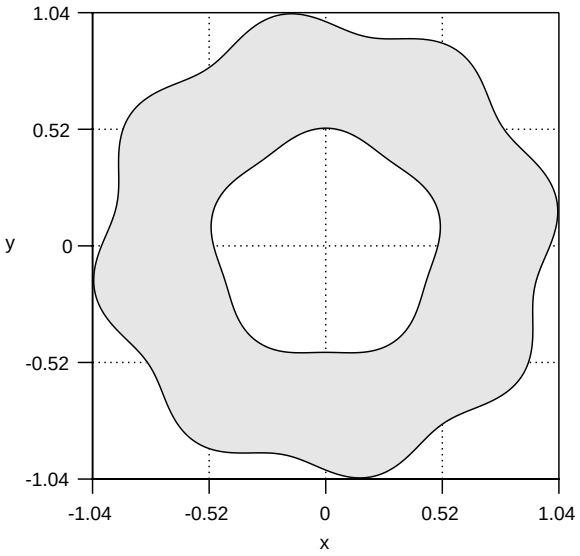
$$\boldsymbol{n}_\mathrm{I}(\theta) = \frac{1}{(R_\mathrm{I}(\theta))^2 + (\partial_\theta R_\mathrm{I}(\theta))^2} \begin{bmatrix} -R_\mathrm{I}(\theta)\cos(\theta) - \partial_\theta R_\mathrm{I}(\theta)\sin(\theta) \\ -R_\mathrm{I}(\theta)\sin(\theta) + \partial_\theta R_\mathrm{I}(\theta)\cos(\theta) \end{bmatrix}, \tag{2.61}$$

$$\boldsymbol{n}_\mathrm{E}(\theta) = \frac{-1}{(R_\mathrm{E}(\theta))^2 + (\partial_\theta R_\mathrm{E}(\theta))^2} \begin{bmatrix} -R_\mathrm{E}(\theta)\cos(\theta) - \partial_\theta R_\mathrm{E}(\theta)\sin(\theta) \\ -R_\mathrm{E}(\theta)\sin(\theta) + \partial_\theta R_\mathrm{E}(\theta)\cos(\theta) \end{bmatrix}, \tag{2.62}$$

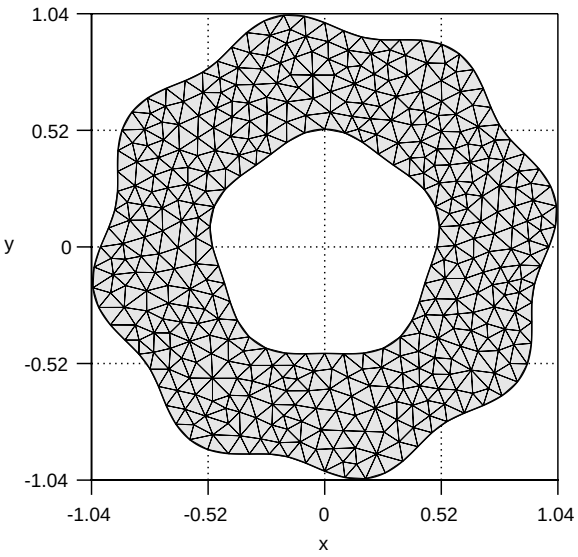
where $\partial_\theta R_\mathrm{I}(\theta)$ and $\partial_\theta R_\mathrm{E}(\theta)$ are the derivatives of $R_\mathrm{I}(\theta)$ and $R_\mathrm{E}(\theta)$, respectively, in order to variable θ.

The simulations are performed with successive finer uniform Delaunay triangular or uniform structured quadrilateral meshes generated for physical domain Ω (see Figure 2.12). The vertices of the boundary edges coincide with the physical boundaries and all the cells in both meshes are

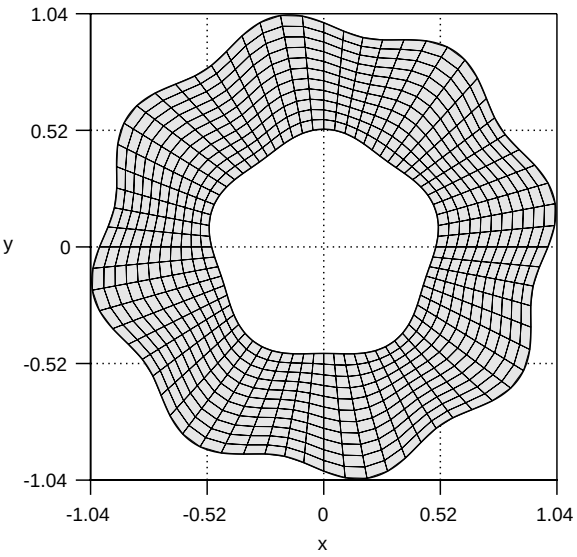
polygonal such that the maximum gap between the physical and the computational boundaries has magnitude order $\mathcal{O}(h^2)$, with h the characteristic mesh size.



(a) Rose-shaped domain.



(b) Uniform Delaunay triangular mesh.



(c) Uniform structured quadrilateral mesh.

Figure 2.12: Domain and coarse meshes for the rose-shaped domain test case.

For this domain, the manufactured solution (see Figure 2.13) is expressed in terms of polar coordinates (r, θ) as

$$\phi(r,\theta) = a(\exp(R(r,\theta)) + \exp(-R(r,\theta)) + b) + 1, \quad \text{in } \Omega, \tag{2.63}$$

$$R(r,\theta) = \frac{2r - (R_{\mathrm{E}}(\theta) + r_{\mathrm{I}})}{R_{\mathrm{E}}(\theta) - r_{\mathrm{I}}}, \quad \text{in } \Omega, \tag{2.64}$$

where $r^2 = x^2 + y^2$ such that $R \in [-1, 1]$, and coefficients $a, b \in \mathbb{R}$. Notice that, although the solution

is given in terms of r and θ, the problem is numerically solved using the Cartesian coordinates. In order to guarantee that $\phi(\boldsymbol{x}) \in [1,2]$ in Ω, coefficients a and b are devised as

$$a = 1/(2 - \exp(1) - \exp(-1)), \tag{2.65}$$

$$b = -\exp(1) - \exp(-1) \tag{2.66}$$

The associated source term function is obtained after substituting manufactured solution (2.63) into Equation (2.1) (see Figure 2.14).

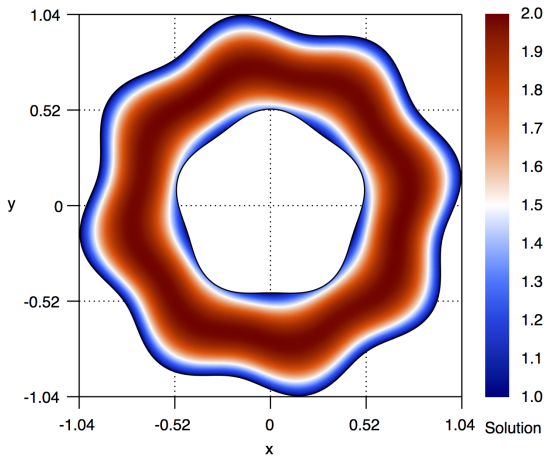


Figure 2.13: Analytic solution for the rose-shaped domain test case.

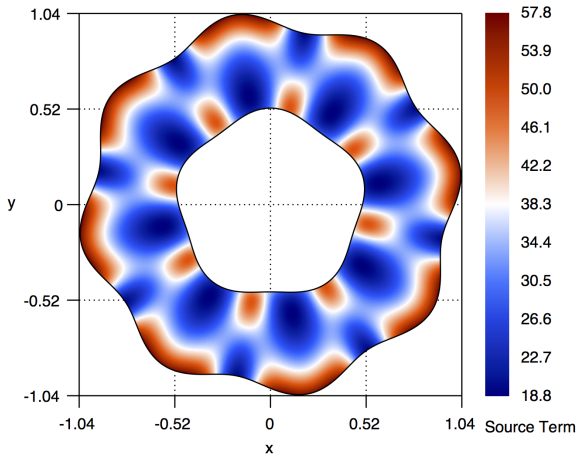


Figure 2.14: Source term for the rose-shaped domain test case.

A convection-diffusion case is considered setting a constant thermal conductivity function $\kappa(r,\theta) = 1$ and a constant radial velocity field $\boldsymbol{u}(r,\theta) = (\cos(\theta), \sin(\theta))$. Internal boundary Γ^{I} is prescribed with a non-constant Dirichlet boundary condition, while external boundary Γ^{E} is prescribed with one of the following cases:

- Dirichlet-Dirichlet boundary conditions: a constant Dirichlet boundary condition.

- Dirichlet-Neumann boundary conditions: a non-constant Neumann boundary condition.

- Dirichlet-Robin boundary conditions: a non-constant Robin boundary condition with functions

$$\alpha^{\mathrm{R}}(r,\theta) = (\theta+\pi)(\theta-\pi)/\pi^2, \quad \text{on } \Gamma^{\mathrm{E}}, \tag{2.67}$$

$$\beta^{\mathrm{R}}(r,\theta) = 1-\alpha^{\mathrm{R}}(\theta), \quad \text{on } \Gamma^{\mathrm{E}}. \tag{2.68}$$

In the previous cases, the associated boundary conditions functions are determined from the analytic solution resulting in cumbersome expressions, which are not provided for the sake of compactness.

2.6.2.1 Dirichlet-Dirichlet boundary conditions

The test case is firstly addressed with Dirichlet-Dirichlet boundary conditions, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, and the ROD and AROD methods with $\mathbb{P}_d/\mathbb{P}_{d+1}$ are carried out. Table 2.9 reports the errors and corresponding convergence orders. As observed, the optimal convergence orders are obtained with this complex geometry, and the ROD and AROD methods perform comparably in terms of accuracy.

The AROD-DN and AROD-DR methods are assessed considering the least-squares constraint, applied to compute the polynomial reconstruction for the boundary edges, of Neumann and Robin types, respectively. At the same time, the fitting condition remains of Dirichlet type. Only the boundary edges associated with the external boundary are treated with the AROD-DN or AROD-DR methods. In contrast, those associated with the internal boundary are treated with the AROD method. When the AROD-DN and AROD-DR methods are used, no GMRES residual convergence was obtained with the last two meshes and polynomial degree $d = 5$ for the applied stopping criteria. On the other side, for the cases where residual convergence is achieved, the AROD-DN and AROD-DR methods provide similar accuracy to the ROD and AROD methods.

To deeper investigate this issue, Table 2.10 reports the number of GMRES iterations in the ROD, AROD, AROD-DN, and AROD-DR methods. A comparable number of GMRES iterations between the ROD and AROD methods is always observed regardless of the polynomial degree and mesh used. However, when the polynomial degree increases and the characteristic mesh size decreases, the AROD-DN and AROD-DR methods provide a larger number of iterations. This behaviour is possibly due to the gradient contribution in the least-squares constraint of Neumann or Robin type in the polynomial reconstruction for each boundary edge, which is used to fulfil a fitting condition of Dirichlet type. Such choice seems to increase the condition number of the system of linear equations, slowing down or even destroying the residual convergence.

Table 2.9: Relative errors and convergence orders obtained in the rose-shaped domain test case with Dirichlet-Dirichlet boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
ROD method with $\mathbb{P}_d$												
750	7.18E−03	—	1.31E−02	—	7.61E−05	—	4.95E−04	—	3.64E−05	—	3.00E−04	—
3 144	1.85E−03	1.89	3.76E−03	1.75	5.03E−06	3.79	3.55E−05	3.68	9.28E−07	5.12	1.62E−05	4.07
12 482	4.77E−04	1.97	1.15E−03	1.71	2.94E−07	4.12	3.46E−06	3.38	1.60E−08	5.89	4.04E−07	5.36
50 102	1.15E−04	2.05	3.03E−04	1.92	2.05E−08	3.84	2.71E−07	3.66	2.85E−10	5.80	1.22E−08	5.03
199 636	2.91E−05	1.99	8.68E−05	1.81	1.28E−09	4.01	1.88E−08	3.86	3.89E−12	6.21	1.75E−10	6.14
AROD method with $\mathbb{P}_d$												
750	7.19E−03	—	1.32E−02	—	7.51E−05	—	4.94E−04	—	3.57E−05	—	3.00E−04	—
3 144	1.85E−03	1.89	3.77E−03	1.75	5.02E−06	3.78	3.51E−05	3.69	9.20E−07	5.10	1.61E−05	4.09
12 482	4.77E−04	1.97	1.16E−03	1.71	2.94E−07	4.12	3.46E−06	3.36	1.60E−08	5.87	4.04E−07	5.34
50 102	1.15E−04	2.05	3.03E−04	1.93	2.05E−08	3.83	2.71E−07	3.66	2.85E−10	5.80	1.22E−08	5.04
199 636	2.91E−05	1.99	8.68E−05	1.81	1.28E−09	4.00	1.88E−08	3.86	3.93E−12	6.20	1.75E−10	6.14
AROD-DN method with $\mathbb{P}_d$												
750	6.58E−03	—	1.24E−02	—	6.97E−05	—	4.79E−04	—	3.48E−05	—	2.91E−04	—
3 144	1.67E−03	1.92	3.49E−03	1.76	4.92E−06	3.70	3.29E−05	3.74	8.84E−07	5.13	1.52E−05	4.11
12 482	4.31E−04	1.96	1.07E−03	1.72	2.81E−07	4.15	3.27E−06	3.35	1.67E−08	5.76	4.28E−07	5.18
50 102	1.03E−04	2.06	2.71E−04	1.97	1.97E−08	3.82	2.57E−07	3.66	—	—	—	—
199 636	2.63E−05	1.98	7.74E−05	1.82	1.27E−09	3.97	1.91E−08	3.76	—	—	—	—
AROD-DR method with $\mathbb{P}_d$												
750	6.58E−03	—	1.24E−02	—	6.97E−05	—	4.79E−04	—	3.48E−05	—	2.91E−04	—
3 144	1.67E−03	1.92	3.50E−03	1.76	4.92E−06	3.70	3.29E−05	3.74	8.84E−07	5.13	1.52E−05	4.11
12 482	4.31E−04	1.96	1.07E−03	1.72	2.81E−07	4.15	3.27E−06	3.35	1.67E−08	5.76	4.28E−07	5.18
50 102	1.03E−04	2.06	2.71E−04	1.97	1.97E−08	3.82	2.57E−07	3.66	—	—	—	—
199 636	2.63E−05	1.98	7.74E−05	1.82	1.27E−09	3.97	1.91E−08	3.76	—	—	—	—

Table 2.10: Number of GMRES iterations obtained in the rose-shaped domain test case with Dirichlet-Dirichlet boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	ROD	AROD	AROD -DN	AROD -DR	ROD	AROD	AROD -DN	AROD -DR	ROD	AROD	AROD -DN	AROD -DR
750	80	40	80	80	80	80	120	120	120	80	280	240
3 144	80	80	120	120	120	120	200	200	200	160	400	400
12 482	120	120	200	200	200	200	440	440	440	320	1 360	1 440
50 102	240	240	400	400	440	440	920	920	600	600	>20 000	>20 000
199 636	520	520	800	800	960	960	2 240	2 240	1 200	1 160	>20 000	>20 000

2.6.2.2 Dirichlet-Neumann boundary conditions

The same test case with Dirichlet-Neumann boundary conditions is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, and the ROD and AROD methods with $\mathbb{P}_d/\mathbb{P}_{d+1}$ are carried out. Table 2.11 reports the errors and corresponding convergence orders. As previously observed, the optimal convergence orders are obtained with this complex geometry, and the ROD and AROD methods perform very comparably. Still, for the latter one, no residual convergence was obtained with the last mesh and polynomial degree $d = 5$.

Similarly to the case with Dirichlet-Dirichlet boundary conditions, the assessment of the AROD-ND and AROD-NR methods is performed considering the least-squares constraint, applied to compute the polynomial reconstructions for the boundary edges, of Dirichlet and Robin types, respectively, while the fitting condition is of Neumann type. Only the boundary edges associated with the external boundary are treated with the AROD-ND or AROD-NR methods. In contrast, those associated with the internal boundary are treated with the AROD method for a more rigorous analysis. As for the AROD method, when the AROD-NR method is applied no residual convergence was obtained with the last mesh and polynomial degree $d = 5$ although, for the cases where residual convergence is achieved, similar accuracy is obtained to the ROD and AROD methods. Moreover, the residual convergence is achieved with all meshes and polynomial degrees when the AROD-ND method is used.

In Table 2.12, the number of GMRES iterations are reported. The results confirm that a different number of iterations are required when the polynomial degree increases or the characteristic mesh size decreases. Concerning the AROD-NR method, the number of GMRES iterations to achieve residual convergence is comparable to the results obtained for the AROD method. The number of GMRES iterations in the AROD-ND method are always comparable to the ROD method ones and smaller than those for the AROD and AROD-NR methods. In conclusion, considering a least-squares constraint of Dirichlet type instead of Neumann or Robin type seems to perform better and provides better conditioning, even when the fitting and the boundary conditions are of Neumann type.

2.6.2.3 Dirichlet-Robin boundary conditions

The same test case with Dirichlet-Robin boundary conditions is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, and the ROD and AROD methods with $\mathbb{P}_d/\mathbb{P}_{d+1}$ are carried out. Table 2.13 reports the errors and corresponding convergence orders. The results are following the ones obtained in the previous Dirichlet-Dirichlet and Dirichlet-Neumann boundary conditions cases, providing the optimal convergence order for the ROD and AROD methods with comparable accuracy.

In Table 2.14, the number of GMRES iterations are reported. As observed before, when the polynomial degree increases or the characteristic mesh size decreases, the AROD-RD method always performs comparably to the ROD method. In contrast, the AROD-RN and AROD methods require more

Table 2.11: Relative errors and convergence orders obtained in the rose-shaped domain test case with Dirichlet-Neumann boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
ROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.33E−03	—	8.50E−03	—	9.41E−05	—	5.36E−04	—	7.51E−05	—	5.35E−04	—
3 144	9.34E−04	1.77	2.72E−03	1.59	5.29E−06	4.02	3.66E−05	3.74	1.73E−06	5.27	1.63E−05	4.87
12 482	2.31E−04	2.02	8.20E−04	1.74	3.83E−07	3.81	3.44E−06	3.43	2.58E−08	6.10	4.04E−07	5.36
50 102	5.78E−05	2.00	2.28E−04	1.84	3.98E−08	3.26	2.72E−07	3.65	5.07E−10	5.66	1.22E−08	5.04
199 636	1.52E−05	1.93	8.29E−05	1.46	3.00E−09	3.74	1.88E−08	3.86	6.99E−12	6.20	1.75E−10	6.14
AROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.30E−03	—	8.40E−03	—	9.38E−05	—	5.34E−04	—	7.57E−05	—	5.40E−04	—
3 144	9.30E−04	1.77	2.71E−03	1.58	5.29E−06	4.01	3.65E−05	3.74	1.72E−06	5.28	1.62E−05	4.90
12 482	2.31E−04	2.02	8.20E−04	1.73	3.82E−07	3.81	3.45E−06	3.42	2.59E−08	6.09	4.05E−07	5.35
50 102	5.77E−05	2.00	2.27E−04	1.85	3.98E−08	3.25	2.72E−07	3.66	5.06E−10	5.66	1.22E−08	5.04
199 636	1.52E−05	1.93	8.28E−05	1.46	3.01E−09	3.73	1.88E−08	3.86	—	—	—	—
AROD-ND method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.31E−03	—	8.40E−03	—	1.04E−04	—	5.83E−04	—	7.76E−05	—	5.37E−04	—
3 144	9.34E−04	1.77	2.71E−03	1.58	5.83E−06	4.02	4.02E−05	3.73	1.74E−06	5.30	1.62E−05	4.89
12 482	2.32E−04	2.02	8.20E−04	1.73	3.57E−07	4.05	3.44E−06	3.57	2.53E−08	6.13	4.05E−07	5.35
50 102	5.80E−05	1.99	2.27E−04	1.85	2.45E−08	3.86	2.71E−07	3.66	5.20E−10	5.59	1.22E−08	5.04
199 636	1.53E−05	1.93	8.29E−05	1.46	1.66E−09	3.89	1.88E−08	3.86	6.82E−12	6.27	1.75E−10	6.14
AROD-NR method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.30E−03	—	8.40E−03	—	9.38E−05	—	5.34E−04	—	7.57E−05	—	5.40E−04	—
3 144	9.30E−04	1.77	2.71E−03	1.58	5.29E−06	4.01	3.65E−05	3.74	1.72E−06	5.28	1.62E−05	4.90
12 482	2.31E−04	2.02	8.20E−04	1.73	3.82E−07	3.81	3.45E−06	3.42	2.59E−08	6.09	4.05E−07	5.35
50 102	5.77E−05	2.00	2.27E−04	1.85	3.98E−08	3.25	2.72E−07	3.66	5.06E−10	5.66	1.22E−08	5.04
199 636	1.52E−05	1.93	8.28E−05	1.46	3.01E−09	3.73	1.88E−08	3.86	—	—	—	—

Table 2.12: Number of GMRES iterations obtained in the rose-shaped domain test case with Dirichlet-Neumann boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	ROD	AROD	AROD -ND	AROD -NR	ROD	AROD	AROD -ND	AROD -NR	ROD	AROD	AROD -ND	AROD -NR
750	80	80	80	80	120	80	120	80	160	200	200	200
3 144	120	120	120	120	240	200	200	200	280	280	320	280
12 482	240	240	240	240	360	360	400	360	720	920	640	920
50 102	440	440	440	440	800	840	800	840	1 480	4 520	1 240	4 640
199 636	840	840	800	840	1 880	1 960	1 800	1 960	5 600	>20 000	3 240	>20 000

GMRES iterations to achieve the same residual convergence.

To conclude the analysis, the ROD, AROD, and AROD-XY methods seem to provide comparable accuracy. However, differences in terms of residual convergence are reported when the polynomial degree increases or the characteristic mesh size decreases. For Dirichlet boundary conditions, the GMRES method with the ROD and AROD methods perform better than the AROD-DN and AROD-DR methods, which require, in general, a larger number of iterations to converge. For Neumann and Robin boundary conditions, the ROD and AROD-ND/AROD-RD methods perform better than the AROD and AROD-NR/AROD-RN methods, which also require, in general, more iterations to converge. Based on these results, the ROD method provides globally the first or second best performance in terms of residual convergence, regardless of the prescribed boundary condition. On the other side, the AROD and AROD-XY methods provide better convergence when the least-squares constraint applied to compute the polynomial reconstructions for the boundary edges is of Dirichlet type, regardless of the type of the boundary condition. In other words, the ROD method for all the cases, and the AROD, AROD-ND, and AROD-RD for the case of a Dirichlet, Neumann, and Robin boundary conditions, respectively, perform better than the others.

2.6.2.4 Quadrilateral meshes

The same test case with Dirichlet-Dirichlet, Dirichlet-Neumann, and Dirichlet-Robin boundary conditions is addressed with successive finer uniform structured quadrilateral meshes. Similarly, the simulations are carried out with polynomial degrees $d = 1, 3, 5$ and the ROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$ polynomial reconstructions. The results for the remaining methods are not reported for the sake of compactness, but similar conclusions are obtained, when compared to the ROD method.

The obtained errors and corresponding convergence orders are reported in Table 2.15. The results support the capability of the method to provide the optimal convergence order with quadrilateral meshes and without any evidence of spurious oscillatory behaviour.

2.6.2.5 Physical boundary heat fluxes

The benchmark section concludes with the analysis of the numerical convective and conductive heat fluxes on the curved physical boundary. Since the numerical heat fluxes are solely computed on the polygonal edges, this case study assesses the capability of the method to extrapolate/interpolate accurate approximations to the physical quantities on the physical boundaries. For that purpose, based on the approximated mean-values in vector $\boldsymbol{\Phi}^{\star}$, obtained from the previous test case with Dirichlet-Dirichlet, Dirichlet-Neumann, and Dirichlet-Robin boundary conditions, polynomial reconstructions are explicitly computed to evaluate the numerical heat fluxes at points falling on the physical boundary. For the sake of simplicity, heat flux evaluations are performed at collocation points $\boldsymbol{b}_{iF} \in \Gamma$ where the polynomial reconstructions for boundary edges e_{iF} on computational boundary

Table 2.13: Number of GMRES iterations obtained in the rose-shaped domain test case with Dirichlet-Robin boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
ROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.50E−03	—	8.55E−03	—	8.71E−05	—	4.95E−04	—	6.89E−05	—	4.86E−04	—
3 144	9.64E−04	1.80	2.72E−03	1.60	5.00E−06	3.99	3.50E−05	3.70	1.60E−06	5.25	1.63E−05	4.74
12 482	2.38E−04	2.03	8.21E−04	1.74	3.69E−07	3.78	3.44E−06	3.36	2.44E−08	6.07	4.04E−07	5.36
50 102	5.87E−05	2.02	2.28E−04	1.84	3.68E−08	3.32	2.72E−07	3.66	4.67E−10	5.70	1.22E−08	5.04
199 636	1.54E−05	1.94	8.15E−05	1.49	2.71E−09	3.77	1.88E−08	3.86	1.63E−11	4.85	1.75E−10	6.14
AROD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.47E−03	—	8.45E−03	—	8.65E−05	—	4.93E−04	—	6.93E−05	—	4.91E−04	—
3 144	9.60E−04	1.79	2.71E−03	1.59	5.00E−06	3.98	3.49E−05	3.69	1.59E−06	5.27	1.62E−05	4.76
12 482	2.38E−04	2.02	8.21E−04	1.73	3.68E−07	3.78	3.45E−06	3.36	2.45E−08	6.06	4.05E−07	5.35
50 102	5.86E−05	2.02	2.28E−04	1.85	3.67E−08	3.32	2.72E−07	3.66	4.65E−10	5.70	1.22E−08	5.04
199 636	1.50E−05	1.97	8.13E−05	1.49	2.70E−09	3.77	1.88E−08	3.86	—	—	—	—
AROD-RD method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.47E−03	—	8.45E−03	—	9.48E−05	—	5.37E−04	—	7.08E−05	—	4.88E−04	—
3 144	9.61E−04	1.79	2.71E−03	1.59	5.37E−06	4.01	3.83E−05	3.69	1.61E−06	5.28	1.61E−05	4.76
12 482	2.38E−04	2.02	8.21E−04	1.73	3.45E−07	3.98	3.44E−06	3.50	2.41E−08	6.10	4.05E−07	5.35
50 102	5.87E−05	2.02	2.28E−04	1.85	2.38E−08	3.85	2.71E−07	3.66	4.77E−10	5.64	1.22E−08	5.04
199 636	1.51E−05	1.97	8.15E−05	1.49	1.58E−09	3.93	1.88E−08	3.86	6.31E−12	6.26	1.75E−10	6.14
AROD-RN method with $\mathbb{P}_d/\mathbb{P}_{d+1}$												
750	3.47E−03	—	8.45E−03	—	8.64E−05	—	4.92E−04	—	6.93E−05	—	4.91E−04	—
3 144	9.60E−04	1.79	2.71E−03	1.59	5.00E−06	3.98	3.49E−05	3.69	1.59E−06	5.27	1.62E−05	4.76
12 482	2.38E−04	2.02	8.21E−04	1.73	3.68E−07	3.78	3.45E−06	3.36	2.45E−08	6.06	4.05E−07	5.35
50 102	5.86E−05	2.02	2.28E−04	1.85	3.67E−08	3.32	2.72E−07	3.66	4.66E−10	5.70	1.22E−08	5.04
199 636	1.50E−05	1.97	8.13E−05	1.49	2.70E−09	3.77	1.88E−08	3.86	—	—	—	—

Table 2.14: Number of GMRES iterations obtained in the rose-shaped domain test case with Dirichlet-Robin boundary conditions.

	$d=1$				$d=3$				$d=5$			
DOF	ROD	AROD	AROD -RD	AROD -RN	ROD	AROD	AROD -RD	AROD -RN	ROD	AROD	AROD -RD	AROD -RN
750	80	80	80	80	80	80	120	80	160	200	160	200
3 144	200	120	120	120	200	160	200	160	280	280	280	280
12 482	200	200	240	200	320	360	360	360	1 080	840	600	880
50 102	440	400	440	400	680	800	760	800	1 440	4 320	1 200	4 280
199 636	840	960	960	960	1 600	2 040	1 880	2 040	4 640	>20 000	2 840	>20 000

Table 2.15: Relative errors and convergence orders obtained in the rose-shaped domain test case with uniform structured quadrilateral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Dirichlet-Dirichlet case												
748	6.89E−03	–	1.19E−02	–	2.08E−04	–	7.85E−04	–	1.28E−04	–	7.96E−04	–
3 128	1.71E−03	1.94	3.15E−03	1.85	1.77E−05	3.44	7.87E−05	3.22	5.03E−06	4.52	7.17E−05	3.36
12 784	4.24E−04	1.99	7.97E−04	1.95	1.28E−06	3.74	5.41E−06	3.80	9.23E−08	5.68	1.28E−06	5.72
52 224	1.04E−04	1.99	1.98E−04	1.98	8.18E−08	3.90	3.31E−07	3.97	1.33E−09	6.02	1.64E−08	6.19
209 984	2.61E−05	1.99	4.96E−05	1.99	5.05E−09	4.00	2.10E−08	3.96	2.63E−11	5.64	2.66E−10	5.92
Dirichlet-Neumann case												
748	2.27E−03	–	9.47E−03	–	4.79E−04	–	2.78E−03	–	2.70E−04	–	2.05E−03	–
3 128	5.88E−04	1.89	2.47E−03	1.88	6.41E−05	2.81	2.73E−04	3.24	4.34E−05	2.56	1.59E−04	3.58
12 784	1.51E−04	1.93	6.30E−04	1.94	4.76E−06	3.69	2.20E−05	3.58	1.28E−06	5.01	3.47E−06	5.43
52 224	3.79E−05	1.96	1.56E−04	1.98	2.59E−07	4.14	1.91E−06	3.48	5.02E−09	7.87	4.19E−08	6.28
209 984	9.41E−06	2.00	3.99E−05	1.96	1.77E−08	3.86	1.91E−07	3.31	3.43E−11	7.17	6.65E−10	5.95
Dirichlet-Robin case												
748	2.36E−03	–	8.76E−03	–	4.33E−04	–	2.67E−03	–	2.62E−04	–	2.00E−03	–
3 128	6.10E−04	1.89	2.31E−03	1.86	5.59E−05	2.86	2.64E−04	3.23	3.66E−05	2.75	1.47E−04	3.64
12 784	1.56E−04	1.94	5.90E−04	1.94	4.15E−06	3.69	2.10E−05	3.60	1.07E−06	5.02	3.28E−06	5.41
52 224	3.90E−05	1.97	1.48E−04	1.97	2.33E−07	4.09	1.88E−06	3.43	4.39E−09	7.81	4.03E−08	6.25
209 984	9.61E−06	2.01	3.77E−05	1.97	1.68E−08	3.77	1.86E−07	3.32	3.54E−11	6.93	6.55E−10	5.92

subset Γ_Δ^F are constrained. Vectors $\boldsymbol{n}_{iF}$ are the associated outward unit normal vectors to the physical boundary at point $\boldsymbol{b}_{iF}$.

The exact physical convective and conductive heat fluxes at point $\boldsymbol{b}_{iF}$, denoted as C_{iF} and D_{iF}, respectively, are given as

$$C_{iF} = (\boldsymbol{u}(\boldsymbol{b}_{iF}) \cdot \boldsymbol{n}_{iF})\phi(\boldsymbol{b}_{iF}), \tag{2.69}$$

$$D_{iF} = -\kappa(\boldsymbol{b}_{iF})\nabla\phi(\boldsymbol{b}_{iF}) \cdot \boldsymbol{n}_{iF}. \tag{2.70}$$

The corresponding numerical convective and conductive heat fluxes at point $\boldsymbol{b}_{iF}$, denoted as $\mathcal{C}_{iF}$ and $\mathcal{D}_{iF}$, respectively, are given as

$$\mathcal{C}_{iF} = [\boldsymbol{u}(\boldsymbol{b}_{iF}) \cdot \boldsymbol{n}_{iF}]^+ \widehat{\varphi}_i(\boldsymbol{b}_{iF}) + [\boldsymbol{u}(\boldsymbol{b}_{iF}) \cdot \boldsymbol{n}_{iF}]^- \widehat{\varphi}_{iF}(\boldsymbol{b}_{iF}), \tag{2.71}$$

$$\mathcal{D}_{iF} = -\kappa(\boldsymbol{b}_{iF})\nabla\widehat{\varphi}_{iF}(\boldsymbol{b}_{iF}) \cdot \boldsymbol{n}_{iF}. \tag{2.72}$$

The L^1- and L^∞-norm errors, denoted as E_1^k and E_∞^k, for the numerical convective heat fluxes

are determined as

$$E_1(\mathcal{M}) = \frac{1}{N_{\mathrm{B}}} \sum_{i \in \mathcal{I}^F, F \in \{\mathrm{D,N,R}\}} |\mathcal{C}_{iF} - C_{iF}|, \tag{2.73}$$

$$E_\infty(\mathcal{M}) = \max_{i \in \mathcal{I}^F, F \in \{\mathrm{D,N,R}\}} |\mathcal{C}_{iF} - C_{iF}|, \tag{2.74}$$

and for the numerical conductive heat fluxes are determined as

$$E_1(\mathcal{M}) = \frac{1}{N_{\mathrm{B}}} \sum_{i \in \mathcal{I}^F, F \in \{\mathrm{D,N,R}\}} |\mathcal{D}_{iF} - D_{iF}|, \tag{2.75}$$

$$E_\infty(\mathcal{M}) = \max_{i \in \mathcal{I}^F, F \in \{\mathrm{D,N,R}\}} |\mathcal{D}_{iF} - D_{iF}|, \tag{2.76}$$

where N_{B} is the number of evaluation points on the physical boundary.

Convergence orders O_1 and O_∞ for the L^1- and L^∞-norm errors, respectively, between two different meshes, $\mathcal{M}_1$ and $\mathcal{M}_2$, with DOF_1 and DOF_2 degrees of freedom (equal to the number of cells and $DOF_1 \neq DOF_2$), respectively, are determined as in Equations (2.46) and (2.47). The ratio of DOF is used instead of N_{B} to compute the convergence orders since the latter does not represent the characteristic mesh size variation. Only the ROD method with $\mathbb{P}_d$ or $\mathbb{P}_d/\mathbb{P}_{d+1}$, $d \in \{1,3,5\}$, polynomial reconstructions is assessed, whereas the AROD and AROD-XY methods provide comparable results.

Remark 8 For some cases, numerical convective heat flux $\mathcal{C}_{iF}$ and/or numerical conductive heat flux $\mathcal{D}_{iF}$ at point $\boldsymbol{b}_{iF}$ are exact and, therefore, are not taken into account to compute either E_1 or E_∞. In particular, such situations arise when polynomial reconstruction $\widehat{\varphi}_{iF}$, whose collocation point is $\boldsymbol{b}_{iF}$, is used to compute the convective or the conductive heat flux when a Dirichlet or Neumann boundary conditions are prescribed at that point, respectively. More specifically, it happens for: Dirichlet-Dirichlet boundary conditions, where the evaluated convective heat fluxes are exact on the internal boundary; Dirichlet-Neumann boundary conditions, where the evaluated convective heat fluxes are exact on the internal boundary and the evaluated conductive heat fluxes are exact on the external boundary; Dirichlet-Robin boundary conditions, where the evaluated convective heat fluxes are exact on the internal boundary

Table 2.16 reports the errors and the corresponding convergence orders for the convective heat fluxes evaluation with Dirichlet-Dirichlet, Dirichlet-Neumann, and Dirichlet-Robin boundary conditions and uniform Delaunay triangular meshes. The results provide numerical evidence that the ROD method recovers convective heat fluxes on the physical boundary and effectively achieves a second-, fourth-, and sixth-order of convergence for polynomial degrees $d = 1,3,5$, respectively, regardless the prescribed boundary condition.

Table 2.17 reports the errors and the corresponding convergence orders for the conductive heat

Table 2.16: Errors and convergence orders for the convective heat fluxes on the physical boundary obtained in the rose-shaped domain test case.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Dirichlet-Dirichlet boundary conditions												
750	5.07E−02	–	9.90E−02	–	1.18E−03	–	4.66E−03	–	8.14E−04	–	3.96E−03	–
3 144	1.33E−02	1.86	2.61E−02	1.86	8.18E−05	3.72	2.72E−04	3.97	1.56E−05	5.52	7.09E−05	5.61
12 482	3.57E−03	1.91	8.34E−03	1.66	6.61E−06	3.65	3.39E−05	3.02	3.00E−07	5.73	2.14E−06	5.08
50 102	9.07E−04	1.97	1.87E−03	2.15	4.12E−07	3.99	2.64E−06	3.68	4.67E−09	5.99	3.52E−08	5.91
199 636	2.26E−04	2.01	4.75E−04	1.98	2.80E−08	3.89	1.72E−07	3.95	8.38E−11	5.82	8.55E−10	5.38
Dirichlet-Neumann boundary conditions												
750	2.53E−02	–	6.36E−02	–	1.57E−03	–	4.98E−03	–	9.17E−04	–	3.45E−03	–
3 144	6.89E−03	1.82	1.80E−02	1.76	1.09E−04	3.72	4.46E−04	3.37	1.91E−05	5.41	8.47E−05	5.17
12 482	1.96E−03	1.82	6.54E−03	1.47	8.28E−06	3.74	3.98E−05	3.50	3.31E−07	5.88	1.80E−06	5.59
50 102	5.37E−04	1.86	1.47E−03	2.15	5.29E−07	3.96	2.45E−06	4.01	5.65E−09	5.86	3.33E−08	5.74
199 636	1.42E−04	1.93	4.09E−04	1.85	3.71E−08	3.85	1.84E−07	3.75	1.27E−10	5.49	7.33E−10	5.52
Dirichlet-Robin boundary conditions												
750	2.40E−02	–	6.09E−02	–	1.61E−03	–	5.05E−03	–	9.48E−04	–	3.55E−03	–
3 144	6.58E−03	1.80	1.79E−02	1.71	1.11E−04	3.73	4.63E−04	3.34	1.97E−05	5.40	8.68E−05	5.18
12 482	1.90E−03	1.80	6.46E−03	1.48	8.35E−06	3.75	3.97E−05	3.56	3.33E−07	5.92	1.78E−06	5.64
50 102	5.30E−04	1.83	1.46E−03	2.14	5.44E−07	3.93	2.44E−06	4.02	5.85E−09	5.82	3.38E−08	5.71
199 636	1.41E−04	1.92	4.08E−04	1.84	3.90E−08	3.81	1.83E−07	3.75	1.56E−10	5.25	8.26E−10	5.37

fluxes evaluation with Dirichlet-Dirichlet, Dirichlet-Neumann, and Dirichlet-Robin boundary conditions and uniform Delaunay triangular meshes. First-, third-, and fifth-order of convergence are achieved for polynomial degrees $d = 1, 3, 5$, respectively, regardless of the prescribed boundary condition. As expected, the convergence orders are one order lower than those obtained for the convective heat fluxes since polynomial reconstructions gradients are computed explicitly to evaluate the conductive heat fluxes.

This analysis shows that the ROD method provides accurate heat fluxes approximations on the curved physical boundary, in consistency with the scheme convergence order. Hence, the method is capable of computing physical quantities, such as mass flow and energy heat flux, with very high-order accuracy performing numerical integration on the physical boundary.

2.7 Conclusions

A very high-order finite volume scheme for the steady-state two-dimensional convection-diffusion problem with curved boundaries has been constructed based on a new framework to compute polynomial reconstructions. The scheme is equipped with the ROD method to handle arbitrary smooth

Table 2.17: Errors and convergence orders for the conductive heat fluxes on the physical boundary obtained in the rose-shaped domain test case.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Dirichlet-Dirichlet boundary conditions												
750	9.89E−02	—	2.63E−01	—	1.36E−02	—	7.48E−02	—	8.38E−03	—	5.36E−02	—
3 144	3.61E−02	1.41	2.12E−01	0.30	1.97E−03	2.70	1.06E−02	2.73	4.88E−04	3.97	3.63E−03	3.76
12 482	1.60E−02	1.18	9.47E−02	1.17	3.00E−04	2.73	2.35E−03	2.18	2.60E−05	4.25	2.71E−04	3.76
50 102	7.19E−03	1.15	5.03E−02	0.91	3.85E−05	2.95	3.31E−04	2.82	9.77E−07	4.73	1.19E−05	4.50
199 636	3.67E−03	0.97	2.90E−02	0.80	5.05E−06	2.94	3.90E−05	3.09	3.44E−08	4.84	4.15E−07	4.86
Dirichlet-Neumann boundary conditions												
750	8.49E−02	—	2.88E−01	—	2.21E−02	—	7.67E−02	—	1.48E−02	—	5.20E−02	—
3 144	3.70E−02	1.16	1.15E−01	1.29	3.81E−03	2.46	1.07E−02	2.75	1.13E−03	3.59	3.62E−03	3.72
12 482	1.35E−02	1.46	5.86E−02	0.97	5.54E−04	2.80	2.35E−03	2.20	6.57E−05	4.13	2.71E−04	3.76
50 102	6.16E−03	1.13	5.13E−02	0.19	7.20E−05	2.94	3.31E−04	2.82	2.52E−06	4.69	1.19E−05	4.50
199 636	3.08E−03	1.00	2.67E−02	0.95	9.64E−06	2.91	3.90E−05	3.09	8.91E−08	4.83	4.16E−07	4.86
Dirichlet-Robin boundary conditions												
750	3.23E−02	—	2.92E−01	—	7.63E−03	—	7.70E−02	—	5.11E−03	—	5.18E−02	—
3 144	1.31E−02	1.26	1.15E−01	1.29	1.28E−03	2.49	1.07E−02	2.75	3.80E−04	3.62	3.62E−03	3.72
12 482	4.70E−03	1.49	5.88E−02	0.98	1.86E−04	2.81	2.35E−03	2.20	2.19E−05	4.14	2.71E−04	3.76
50 102	2.10E−03	1.16	5.13E−02	0.20	2.41E−05	2.94	3.31E−04	2.82	8.40E−07	4.69	1.19E−05	4.50
199 636	1.04E−03	1.01	2.67E−02	0.95	3.22E−06	2.91	3.90E−05	3.09	2.97E−08	4.83	4.16E−07	4.86

curved boundaries prescribed with general boundary conditions, such as the Dirichlet, Neumann, and Robin types. The method does not require curved elements to fit the boundary, nor complex quadrature rules for integration, nor complex non-linear transformations, which represents a significant achievement to the existing techniques in the finite volume context. The AROD method and its derivations were also proposed aiming at evolving boundaries and interfaces in time or another kind of iterative process. This approach avoids recomputing the least-squares procedure for small boundary displacements, ultimately saving computational resources. A comprehensive numerical benchmark test suit, using the method of manufactured solutions, verified the implementation and assessed the performance of the proposed methods with simple and complex curved shaped domains.

Very high accurate convergence orders were achieved for the two-dimensional steady-state pure convection, pure diffusion, and convection-diffusion problems, either with triangular and quadrilateral meshes. To achieve the optimal convergence orders when Neumann and Robin boundary conditions are prescribed, polynomial degree $d+1$ is required for the associated boundary edges while polynomial degree d can be used for the remaining polynomial reconstructions for the boundary and inner edges and cells. Since polynomial degree d is required to treat Dirichlet boundary conditions, a performance penalty is expected when treating Neumann and Robin boundary conditions. In practice, such penalty

is relatively small as it only affects the polynomial reconstructions associated with the Neumann or Robin boundary edges. For the three-dimensional case, such penalty would be negligible since the ratio of the number of boundary faces by the number of inner faces and cells decreases, when compared with the two-dimensional case. Both the ROD, AROD, and AROD-XY methods provide comparable accuracy and the optimal convergence orders when handling boundary conditions on curved boundaries, while no non-physical oscillations are obtained for smooth solutions. Using a polynomial degree d for the Dirichlet boundary edges and cells, and polynomial degree $d+1$ for the Neumann and Robin boundary edges, a $(d+1)$th-order of convergence is obtained. Although all the methods provide comparable accuracy, a performance deterioration in terms of GMRES convergence occurs in some situations, particularly when increasing the polynomial degree or refining the mesh. To summarize, the ROD method, regardless of the boundary condition, and the AROD, AROD-ND, and AROD-RD methods, for Dirichlet, Neumann, and Robin boundary conditions, respectively, provide the best performance.

The plans within this research include to apply the proposed ROD, AROD, and AROD-XY methods to complex system models, such as the Cauchy, Navier-Stokes, and Euler equations, in three-dimensional geometries with structured and unstructured meshes. Additionally, unsteady problems with time-evolving domains, such as piston, pulsating interfaces, and tracking interfaces problems, will be investigated aiming towards complex physical and industrial applications. Further verification of these complex problems and applications will be required.

The method has only been tested with regular solutions for the scalar linear convection-diffusion equation. In contrast, there is no guarantee that the method is suitable for complex situations, such as catching boundary layers or turbulent flows in the vicinity of curved boundaries or interfaces, preserving the mass conservation in divergence-free flows or multiphase flows. Many questions remain open. Nevertheless, meshing with curved elements introduces additional complexity and computational cost. Hence, the possibility to avoid such kind of elements, at least, for simple situations, is a contribution to alleviate the computational effort and to simplify the existing numerical schemes with straightforward quadrature rules, more simple flux expressions, conventional algorithms for polygonal meshes, among others.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] P.-H. Cournede, C. Debiez, A. Dervieux, A positive MUSCL scheme for triangulations, INRIA Report 3465 (1998).

[11] T.J. Barth, P.O. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper 90-0013 (1990).

[12] T.J. Barth, Recent developments in high order k-exact reconstruction on unstructured meshes, AIAA Paper 93-0668 (1993).

[13] B. Leer, Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme, J. Comput. Phys. 14(4) (1974) 361–370.

[14] V. Venkatakrishnan, On the accuracy of limiters and convergence to steady-state solutions, AIAA paper 93-0880 (1993).

[15] M.E. Hubbard, Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids, J. Comput. Phys. 155 (1999) 54–74.

[16] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.

[17] D. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, J. Comput. Phys. 160 (2000) 405–452.

[18] M. Dumbser, M. Käser, Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, J. Comput. Phys. 221 (2007) 693–723.

[19] C.-W. Shu, High-order weighted essentially non-oscillatory schemes for convection dominated problems, SIAM Review 51 (2009) 82–126.

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

[21] J. Qiu, C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, SIAM J. Sci. Comput. 26 (2005) 907–929.

[22] R. Abgrall, T. Sonar, On the use of Mühlbach expansions in the recovery step of ENO methods. Numerische Mathematik 76 (1997) 1–25.

[23] P. Colella, M.D. Sekora, A limiter for PPM that preserves accuracy at smooth extrema. J. Comput. Phys. 227 (2008) 7069–7076.

[24] D. Kuzmin, A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods, J. Comput. Appl. Math. 233(12) (2010) 3077–3085.

[25] S. Clain, G.J. Machado, J.M. Nóbrega, R.M.S. Pereira, A sixth-order finite volume method for the convection-diffusion problem with discontinuous coefficients, Comput. Meth. Appl. Mech. Engrg. 267 (2013) 43–64.

[26] T.J. Barth, Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations, AGARD Report 787 (1992).

[27] A. Boularas, S. Clain, F. Baudoin, A sixth-order finite volume method for diffusion problem with curved boundaries, Appl. Math. Model. 42 (2017) 401–422.

[28] D.P. Bertsekas, Constrained optimization and lagrange multiplier methods, 1st Edition, Academic Press (1982).

[29] C. Ollivier-Gooch, A. Nejat, C. Michalak, On obtaining high-order finite-volume solutions to the Euler equations on unstructured meshes, AIAA Paper 2007-4464 (2007).

[30] C. Ollivier-Gooch, A. Nejat, C. Michalak, Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations, AIAA Journal 47(9) (2009) 2105–2120.

[31] C. Michalak, C. Ollivier-Gooch, Unstructured high-order accurate finite volume solutions of the Navier-Stokes equations. AIAA Paper 2009-954 (2009).

[32] A. Ern, J.-L. Guermond, Theory and practice of finite elements, 1st Edition, Springer (2004).

CHAPTER 3

Very high-order accurate polygonal mesh finite volume scheme for conjugate heat transfer problems with curved interfaces and imperfect contacts

Abstract: The conjugate heat transfer problem is found in non-isothermal physical systems that involve thermodynamic processes between materials that are thermally coupled through non-adiabatic contacts. A very high-order accurate finite volume scheme in general polygonal meshes is proposed to solve conjugate heat transfer problems with arbitrary curved interfaces and imperfect thermal contacts. Besides the discontinuous physical properties, imperfect thermal contacts are challenging to address since the obtained temperature is also discontinuous on the interface as a consequence of the interfacial thermal resistance. Moreover, the arbitrary curved interfaces are discretized with polygonal meshes to avoid the shortcomings of the common curved mesh approaches. Still, a specific treatment is required to overcome the geometrical mismatch and adequately fulfil the prescribed interface conditions. The proposed method is based on a partitioned formulation of the conjugate heat transfer problem with the appropriate thermal coupling. A generic polynomial reconstruction method is used to provide local approximations of the temperature complemented with the reconstruction for off-site data method to fulfil the prescribed interface conditions properly. A comprehensive numerical benchmark is provided to verify the proposed method and assess its capability in terms of accuracy, convergence order, stability, and robustness. The results provide the optimal very high-order of convergence and prove the capability of the proposed method to handle arbitrary curved interfaces and imperfect thermal contacts. This contribution represents a significant step towards more efficient and versatile numerical methods for complex conjugate heat transfer problems in engineering applications.

Keywords: Conjugate heat transfer problems, arbitrary curved interfaces, imperfect thermal contacts, general polygonal meshes, reconstruction for off-site data method, very high-order accurate finite volume scheme

This chapter was adapted from R. Costa, J.M. Nóbrega, S. Clain, and G.J. Machado, Very high-order accurate polygonal mesh finite volume scheme for conjugate heat transfer problems with curved interfaces and imperfect contacts, Comput. Meth. Appl. Mech. Engrg. 357 (2019) 112560, DOI: https://doi.org/10.1016/j.cma.2019.07.029

3.1 Introduction

The conjugate heat transfer refers to non-isothermal physical systems that involve thermodynamic processes between solids and fluids, consisting of materials with different physical properties that are thermally coupled through non-adiabatic contacts. The problem of the conjugate heat transfer consists in determining the temperature distribution in these multi-material domains with specific thermodynamic laws applied for the heat transfer on the contacts. Many multiphysics problems in fluid mechanics, solid mechanics, and electromagnetics involve multi-material domains and physical quantities that depend on the temperature, such as thermomechanics, thermoelasticity, electrothermomagnetics, and fluid-thermal-structure interaction [1–4]. Hence, the numerical solution and analysis of conjugate heat transfer problems provide means of investigating and understating the intricate thermodynamics in a wide range of complex applications, from the nuclear to aerospace engineering.

3.1.1 Mathematical modelling of thermal contacts

Conjugate heat transfer problems are formulated as elliptic interface problems in partitioned domains, where the subdomains correspond to the different materials, and the interfaces correspond to the contacts. The model consists of second-order elliptic or parabolic partial differential equations and, besides the usual boundary conditions on the boundaries, specific constraints for the temperature and the conductive heat flux are prescribed on the interfaces and referred to as interface conditions. Additionally, since the materials in contact have different physical properties, the model also consists of a discontinuous thermal conductivity, heat capacity, and density on the interfaces. Additionally, a discontinuous velocity or heat source is also possible. For these reasons, specialized methods with the appropriate treatment for interface conditions and discontinuous physical properties are required to solve conjugate heat transfer problems numerically.

The most straightforward and commonly applied constraints on the interfaces impose the continuity of the temperature in addition to the conservation of the conductive heat flux. They are referred to as the continuity interface conditions or the homogeneous jump interface conditions. Although continuous temperature and conservative conductive heat flux are recovered on these perfect thermal contacts, the temperature derivative normal to the interface is discontinuous due to the conservation of the conductive heat flux and the adjacent materials with different thermal conductivity. The produced finite jump is proportional to the ratio of the thermal conductivity such that the unit case recovers the single-material domain problem with continuous temperature, conductive heat flux, and temperature derivative normal to the interface. Additionally, suppose the contact has an interfacial heat source (usually due to frictional sliding). In that case, an explicit finite jump is considered for the conductive heat flux conservation on the interfaces and corresponds to the generated heat rate per unit area (explicit finite jumps for the temperature can also be considered but are of little practical

interest).

In contrast, for solids sliding relative to one another, an interfacial thermal resistance (also referred to as Kapitza resistance) hampers the heat transfer between materials on the contacts due to the roughness of their imperfect surfaces [5, 6]. Although conservative conductive heat flux is recovered on these imperfect thermal contacts, the temperature and its derivative normal to the interface are discontinuous (other types of imperfect thermal contacts are found in Javili et al., 2014 [7]), for which reason the continuity interface conditions are inappropriate. The produced finite jump is proportional to the interfacial thermal resistance such that the null case recovers the continuity interface conditions. More specifically, the applied constraints on the interfaces impose the conservation of conductive heat flux given as a function of the interfacial thermal resistance and the implicit jump of the temperature (provided no interfacial heat source). They are referred to as the imperfect interface conditions [8–10].

A wide range of emerging engineering applications concerns multi-material problems with a significant interfacial thermal resistance on the contacts. For instance, cooling systems for polymer processing, thermal rectifiers, high-performance thermoelectric materials, and carbon nanotube-electrode-based materials. Nonetheless, although the conjugate heat transfer with imperfect thermal contacts is a comprehensive problem, the majority of the numerical methods are only capable of treating the continuity interface conditions for perfect thermal contacts. In contrast, very few are capable of treating the former. Moreover, conjugate heat transfer problems with complex curved interfaces are also underdeveloped, in particular for very high-order accurate methods since challenging issues arise to preserve the optimal convergence orders. Therefore, there is an increasing need for numerical methods that solve efficiently conjugate heat transfer problems with curved interfaces and imperfect thermal contacts to provide accurate approximate solutions at a reduced computational cost for a wide range of engineering applications. Moreover, other physical problems consisting of physical quantities with implicit jumps on the interface, such as velocity, displacement, and potential, benefit from the numerical developments in that context.

3.1.2 Numerical approaches for conjugate problems

There is a large body of published work concerning the development and application of numerical methods attempting to solve efficiently conjugate heat transfer problems or other conjugate problems with interface conditions for physical quantities such as velocity, displacement, and potential. In general, the numerical approaches are divided into two generic categories, namely monolithic or partitioned [11–23]. However, in multiphysics problems, some physical quantities are computed in a monolithic approach, while a partitioned approach is applied to compute the others.

The monolithic approach (also referred to as the direct approach) consists in using a single tailored method that combines the spatial discretization of the governing equations for all subdomains, including boundary and interface conditions. It produces a single system of linear equations to solve. For instance, a mixture representation of the physical properties is considered in the vicinity of the

interface to provide a continuous and smooth transition from one subdomain to the other. Then, a single governing equation with only boundary conditions for the heat transfer in all subdomains is derived, and the spatial discretization is applied without any separation between materials [24]. However, interface conditions for imperfect thermal contacts require more elaborated techniques, as the conditional volume averaging, which considers additional terms in the governing equation [24–26]. Moreover, several limitations and challenges arise to overcome the usual first- and second-order of convergence provided with the discretization schemes for the monolithic approach [24].

On the contrary, the partitioned approach (also referred to as the segregated approach) consists in a differential partitioning method that decomposes the conjugate heat transfer problem into several individual subproblems of heat transfer (one per subdomain). Moreover, interface conditions are converted into boundary conditions on the interface for each subproblem. The thermal coupling between the subproblems consists in providing the appropriate functions associated with the produced boundary conditions for the data transfer such that the solution of the partitioned problem satisfies the interface conditions and, therefore, corresponds to the solution of the conjugate heat transfer problem. The traditionally applied partitioning method in the case of the continuity interface conditions, referred to as the Dirichlet-Neumann method, prescribes a Dirichlet boundary condition to impose the temperature and a Neumann boundary condition to impose the conductive heat flux. Therefore, apart from the initially prescribed boundary conditions, a Dirichlet boundary value subproblem is solved in one subdomain, and a Neumann boundary value subproblem is solved in the adjacent one. The Dirichlet boundary condition is commonly prescribed in the subproblem with the lower thermal conductivity and the Neumann boundary condition in the other subproblem [17], which usually maximizes the stability of the Dirichlet-Neumann method.

Recently, for unsteady conjugate heat transfer problems, the Dirichlet-Robin or Robin-Robin methods have been investigated as an alternative to the traditional Dirichlet-Neumann method. This approach aims at enhancing the convergence of the partitioned approach but, unfortunately, the optimal coefficients for the Robin boundary condition are not trivial to determine [18–23]. In the case of imperfect interface conditions, the applied partitioning method, referred to as the Neumann-Neumann method, produces two Neumann boundary value subproblems to be solved in both adjacent subdomains.

In comparison with the monolithic approach, the partitioned approach has the advantage of discretizing the subproblems separately. Therefore, different discretization schemes or time-steps for unsteady problems can be applied depending on the physical properties of the materials [4]. In addition to the internal variables associated with the discretized subproblems, boundary variables to approximate the boundary condition functions are also usually necessary. However, a static condensation can be applied to the system of linear equations, reducing the number of variables to determine.

In the case of unsteady conjugate heat transfer problems, the partitioned approach is also classified as strongly- or loosely-coupled (also called weakly-coupled). In the strongly-coupled case,

considering equal time-steps for all subproblems, the method is equipped with an iterative algorithm between each pair of consecutive time-stations intended to solve the subproblems sequentially until some stopping criteria, usually a residual tolerance, is achieved. Roughly speaking, and apart from the spatial discretization considered, the strongly-coupled case is equivalent to the monolithic approach in terms of accuracy and temporal stability. However, in comparison with the monolithic approach, the convergence of iterative algorithms can be very slow, in particular for cases with very distinct thermal conductivity of the adjacent subdomains. In that case, large relaxation coefficients are required to guarantee the stability and convergence of the solution, and coarser grid solvers are often employed to obtain a good initial guess solution. On the other hand, in an attempting to simplify the algorithms and improve computational efficiency, some authors prefer to perform only one subiteration, where the data transfer between subdomains through the boundary conditions occurs only once within each time iteration. The loosely-coupled case comes at the cost of severe degradation in accuracy and stability and, therefore, a careful and often complex and cumbersome numerical analysis of implemented algorithms is crucial to avoid loss in accuracy and convergence orders or even solutions that diverge in time. Moreover, it is susceptible to the partitioning configuration, that is, in which subproblem the Dirichlet or Neumann boundary conditions are prescribed. In that case, the ratio of thermal effusivities (the ability of the material to transfer thermal energy with its surroundings) is usually used as a measure of the thermal interaction strength. Then, the Dirichlet condition is prescribed in the subproblem with the lower thermal effusivity and the Neumann condition in the other subproblem [12–14, 19].

3.1.3 Very high-order accurate methods

The ever-increasing use of computer simulations for more-than-ever complex multiphysics models led to an emerging need of developing accurate, robust, stable, and efficient numerical methods. In that regard, the case of the conjugate heat transfer is of paramount importance to deliver reliable simulations of the thermal processes in engineering applications. Very high-order accurate methods (here defined as more than the second-order of convergence) are unquestionably a useful framework to increase the approximate solution accuracy and to achieve substantial computational gains [27, 28]. Significant developments have been made in the last decades but, unfortunately, obtaining very high-order accurate, robust and stable methods that outperform the efficiency, simplicity, and versatility of traditional ones is still a challenging task [27, 28].

The literature concerning the spatial discretization of conjugate heat transfer problems is extensive and uncountable methods are capable of treating the continuity interface conditions, for instance, finite element methods, finite volume methods, finite difference methods, and Fourier-spectral methods. However, the first- and second-orders of convergence are usually obtained under mesh refinement, whereas few methods are capable to effectively achieve a very high-order of convergence, for instance, the internal penalty discontinuous Galerkin method [29, 30], hybridizable discontinuous Galerkin method [31–33] (also for the Stokes interface flow problem [34]), cut-cell discontinuous Galerkin method [35, 36], weak Galerkin finite element method [37, 38], finite element method [39–45],

finite difference method [46–50], finite cell method [51], among a few others. Contrarily to the case of the continuity interface conditions, very few methods are capable of treating the imperfect interface conditions, for instance, in the context of finite volume methods [52, 53] and finite element methods [54]. Unfortunately, none of these methods can achieve a very high-order of convergence.

3.1.4 Curved boundaries and interfaces treatment

Solving conjugate heat transfer problems consisting of curved domains discretized with polygonal meshes might drastically reduce the convergence order of very high-order accurate methods if the appropriate treatment of the boundary and interface conditions is not provided. Indeed, the geometrical mismatch between the curved boundaries or interfaces and the associated polygonal approximation, with magnitude order $\mathcal{O}(h^2)$, with h the characteristic mesh size, might result in the second-order of convergence at most. Indeed, treating curved domains usually requires more elaborated methods than polygonal domains to preserve the optimal convergence orders. The three-dimensional case, in addition to the generic boundary and interface conditions prescribed on arbitrary curved boundaries and interfaces, usually represent an increased challenge. Therefore, avoid numerical limitations and additional computational costs is of crucial importance to efficiently treat conjugate heat transfer problems in curved domains.

The standard approach to handle curved domains is to avoid the geometrical mismatch using curved mesh elements that follow the curved boundaries and interfaces (inner mesh elements might also be curved for robustness purposes), as the classic isoparametric elements method [55, 56]. However, the use of curved meshes comes at the cost of cumbersome and computationally intensive generation algorithms [27, 28, 57, 58] and complex non-linear transformations for the finite element basis functions. Additionally, very high-order accurate quadrature rules are also necessary for the curved mesh elements, which are very intricate to obtain in the general case and, in particular, for arbitrary curved three-dimensional geometries [59–63].

As these difficulties are the primary outcome of the use of curved meshes, a few methods capable of treating curved boundaries and interfaces with polygonal meshes have recently emerged, as the reconstruction for off-site data (ROD) method proposed in Costa et al., 2018 [64, 65]. The method is based on polynomial reconstructions of a certain degree to provide finite volume and finite differences discretizations, for instance, with very high-order of convergence solely using polygonal meshes generated for arbitrary curved domains. The method is also capable of treating the fundamental Dirichlet and Neumann boundary conditions prescribed on arbitrary curved boundaries. However, the case of the interface conditions prescribed on arbitrary curved interfaces has not been addressed yet. Besides overcoming the drawbacks associated with the use of curved meshes, the ROD method is a very versatile technique, easy to implement, and suitable to solve, with very few limitations, complex problems in two- and three-dimensional arbitrary curved domains.

3.1.5 Novelties of the work

In the present work, a very high-order accurate finite volume method is proposed to solve the steady-state conjugate heat transfer problem in two dimensions. The novelties of the work correspond to the main challenges to preserve the optimal convergence order of the method in this context. Firstly, provide the appropriate partitioning method for the imperfect interface conditions, where the continuity interface conditions are also considered to complement the method. Secondly, provide the proper treatment for arbitrary curved interfaces with the sole use of polygonal meshes and without the main limitations of the common approaches with curved meshes.

In general terms, the proposed method is based on the partitioned approach converting the conjugate heat transfer problem into individual heat transfer subproblems for the subdomains. The Dirichlet-Neumann and Neumann-Neumann methods are considered to provide the partitioned problem model. The appropriate thermal coupling between the subproblems consists in providing the proper boundary condition functions such that the solution of the partitioned problem satisfies the interface conditions and, therefore, corresponds to the solution of the conjugate heat transfer problem. Then, the produced Dirichlet and Neumann boundary value subproblems are discretized in terms of internal and boundary variables with a finite volume method using solely general polygonal meshes. For curved domains, the ROD method is applied to provide the appropriate treatment of the boundary and interface conditions prescribed on the arbitrary curved boundaries and interfaces. Finally, a static condensation of the produced system of linear equations is derived from eliminating the boundary variables, which allows solving the problem only in terms of internal variables.

3.1.6 Organization of the chapter

The remaining sections of the chapter are organized as follows. Section 2 presents the conjugate problem model, the partitioned problem model resulting from the partitioning method, and the thermal coupling between subproblems. Section 3 introduces the polygonal meshes, the polynomial reconstruction method, and the least-squares method. Section 4 prescribes the polynomial reconstructions for the proposed method and the application of the ROD method. Section 5 is dedicated to the generic finite volume discretization, the very high-order accurate scheme, the implicit system of linear equations in terms of affine residual operators, and the associated static condensation. Section 6 provides a comprehensive numerical benchmark based on manufactured solutions for the conjugate heat transfer problem to verify the proposed method. The chapter is completed in Section 7 with the conclusions and some perspectives for future work.

3.2 Mathematical formulation

The steady-state conjugate heat transfer problem is addressed in two dimensions and formulated with the Cartesian coordinate system considering $\boldsymbol{x} := (x, y)$. For the sake of simplicity and without loss of generality, only two subdomains are considered and treated, but the problem is easily formulated in the case of multiple subdomains.

3.2.1 Curved domain

Let Ω denote an open bounded arbitrary curved domain in $\mathbb{R}^2$ with boundary Γ^{D}, partitioned into two non-overlapping subdomains, Ω^S, $S \in \{\mathrm{A}, \mathrm{B}\}$, sharing interface Γ^{I}, as illustrated in Figures 3.1. The boundary and the interface are regular Jordan curves, that is, simple and closed curves, and admit a known local parameterization. Boundary Γ^{D} is partitioned into two non-overlapping subsets (where one of the two can be the empty set as in the case of two concentric subdomains), denoting as $\Gamma^{\mathrm{D},S}$ the boundary of subdomain Ω^S, such that $\Gamma^{\mathrm{D}} = \Gamma^{\mathrm{D,A}} \cup \Gamma^{\mathrm{D,B}}$. Interface Γ^{I} is partitioned into two non-overlapping subsets (where again, one of them can be the empty set), denoted as Γ^{C} and Γ^{K}, such that $\Gamma^{\mathrm{I}} = \Gamma^{\mathrm{C}} \cup \Gamma^{\mathrm{K}}$.

For each subdomain Ω^S, let Γ^S denote the respective boundary, consisting of boundaries and interface subsets of domain Ω, namely $\Gamma^S = \Gamma^{\mathrm{D},S} \cup \Gamma^{\mathrm{I}} = \Gamma^{\mathrm{D},S} \cup \Gamma^{\mathrm{C}} \cup \Gamma^{\mathrm{K}}$. Moreover, for any point $\boldsymbol{x}$ on boundary Γ^S, vector $\boldsymbol{n}^S := \boldsymbol{n}^S(\boldsymbol{x}) := \left(n_x^S(\boldsymbol{x}), n_y^S(\boldsymbol{x})\right)$ stands for the unit normal vector outward to subdomain Ω^S (see Figure 3.1). Notice that if $\boldsymbol{x} \in \Gamma^{\mathrm{I}}$, therefore, $\boldsymbol{n}^{\mathrm{A}}(\boldsymbol{x}) = -\boldsymbol{n}^{\mathrm{B}}(\boldsymbol{x})$.

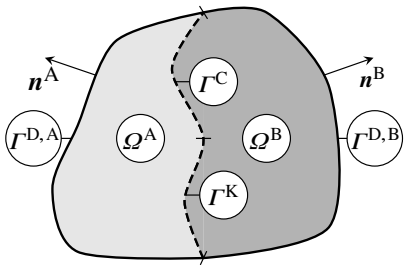


Figure 3.1: Example of arbitrary curved domain with subdomains, boundary subsets, interface subsets, and outward unit normal vectors.

3.2.2 Conjugate problem model

The conjugate heat transfer problem consists in seeking temperature function $\phi^S(\boldsymbol{x})$ in subdomain Ω^S with governing partial differential equation given as

$$\nabla \cdot \left(\boldsymbol{u}^S \phi^S - \kappa^S \nabla \phi^S\right) = f^S, \quad \text{in } \Omega^S, \tag{3.1}$$

where $\boldsymbol{u}^S := \boldsymbol{u}^S(\boldsymbol{x}) := \left(u_x^S(\boldsymbol{x}), u_y^S(\boldsymbol{x})\right)$ is the velocity vector function multiplied by the heat capacity and density of the associated material, $\kappa^S := \kappa^S(\boldsymbol{x})$ is the thermal conductivity function, and $f^S := f^S(\boldsymbol{x})$ is the heat source function (a negative value implies a heat sink) defined in subdomain Ω^S. All functions are assumed regular and bounded in the associated subdomains.

Notice that, in the case of discontinuous temperature, velocity, thermal conductivity, and heat source on the interface, associated functions $\phi^S(\boldsymbol{x})$, $\boldsymbol{u}^S(\boldsymbol{x})$, $\kappa^S(\boldsymbol{x})$, and $f^S(\boldsymbol{x})$ are continuous (also regular and bounded) in the associated subdomains. The term discontinuous refers to the physical quantity (temperature, velocity, thermal conductivity, and heat source) defined in the entire domain. In contrast, the associated functions are defined only in the associated subdomains. Therefore, if a given physical quantity is discontinuous, associated functions defined in each of the subdomains are continuous but do not correspond on the interface.

No mass transfer from one subdomain to the other is considered since in conjugate heat transfer problems it is not physical to have a convective transport through the interface. Therefore, functions $\boldsymbol{u}^S(\boldsymbol{x})$ are always given tangent to interface Γ^{I}, that is, $\boldsymbol{u}^S(\boldsymbol{x}) \cdot \boldsymbol{n}^S(\boldsymbol{x}) = 0$.

Remark 9 The term regular is used to indicate that the function belongs to spaces C^{k+1}, with $k \geq d$, when polynomial reconstruction of degree d is used to achieve a $(d+1)$-order accurate scheme. In other words, the function is continuous and has, at least, continuous first $k+1$ derivatives.

The model problem is complemented with the Dirichlet boundary condition on boundary subsets $\Gamma^{\mathrm{D},S}$, given as

$$\phi^S = g^{\mathrm{D},S}, \quad \text{on } \Gamma^{\mathrm{D},S}, \tag{3.2}$$

where $g^{\mathrm{D},S} := g^{\mathrm{D},S}(\boldsymbol{x})$ are given regular and bounded functions. Although Neumann and Robin boundary conditions can also be prescribed in conjugate heat transfer problems, they are not elaborated in the present work for the sake of simplicity (more details concerning this topic are found in Costa et al., 2018 [65]).

Additionally, the model problem also requires interface conditions and two cases are considered, as follows:

- The continuity interface conditions (or homogeneous jump interface conditions [54]) on interface subset Γ^{C}, given as

$$\left.\begin{aligned} \phi^{\mathrm{A}} &= \phi^{\mathrm{B}}, && (3.3)\\ -\kappa^{\mathrm{A}} \nabla\phi^{\mathrm{A}} \cdot \boldsymbol{n}^{\mathrm{A}} - \kappa^{\mathrm{B}} \nabla\phi^{\mathrm{B}} \cdot \boldsymbol{n}^{\mathrm{B}} &= 0, && (3.4) \end{aligned}\right\} \quad \text{on } \Gamma^{\mathrm{C}},$$

  which imposes the continuity of the temperature and the conservation of the conductive heat flux on the associated interface subset. Notice that, for adjacent materials with different thermal

conductivity, the temperature derivative normal to the interface is necessarily discontinuous in order to conserve the conductive heat flux.

- The imperfect interface conditions (or Robin type jump or Kapitza interface conditions [54]) on interface subset Γ^{K}, given as

$$\left.\begin{aligned} -\kappa^{\mathrm{A}}\nabla\phi^{\mathrm{A}}\cdot\boldsymbol{n}^{\mathrm{A}} &= h^{\mathrm{K}}\Big(\phi^{\mathrm{A}}-\phi^{\mathrm{B}}\Big), \qquad (3.5)\\ -\kappa^{\mathrm{A}}\nabla\phi^{\mathrm{A}}\cdot\boldsymbol{n}^{\mathrm{A}} &= \kappa^{\mathrm{B}}\nabla\phi^{\mathrm{B}}\cdot\boldsymbol{n}^{\mathrm{B}}, \qquad (3.6)\end{aligned}\right\} \quad \text{on } \Gamma^{\mathrm{K}},$$

  where $h^{\mathrm{K}} := h^{\mathrm{K}}(\boldsymbol{x}) = 1/r^{\mathrm{K}}(\boldsymbol{x})$ is a given interfacial thermal conductance function (its inverse is called interfacial thermal resistance coefficient or Kapitza coefficient), which imposes the conservation of the conductive heat flux on the associated interface subset given as function of the unknown finite jump of the temperature. These interface conditions represent a lowly-conductive imperfect interface also known as Kapitza interface [5, 7]. Notice that the temperature is discontinuous on the associated interface subset, where the exceptional case with functions $\phi^{\mathrm{A}}(\boldsymbol{x}) = \phi^{\mathrm{B}}(\boldsymbol{x})$ results in the adiabatic condition (no conductive heat flux). As for the continuity interface condition, the temperature derivative normal to the interface is discontinuous for adjacent materials with different thermal conductivity.

Remark 10 As addressed in Clain et al., 2013 [66], the continuity interface conditions can be recovered assigning a high interfacial thermal conductance function for the imperfect interface conditions. In fact, if $h^K(\boldsymbol{x}) \to \infty$ then $\big(-\kappa^A(\boldsymbol{x})\nabla\phi^A(\boldsymbol{x})\cdot\boldsymbol{n}^A(\boldsymbol{x})\big)/h^K(\boldsymbol{x}) \to 0$ and, therefore, $\phi^A(\boldsymbol{x}) \to \phi^B(\boldsymbol{x})$. In practice, the authors verified that excessively high interfacial thermal conductance functions are required to recover the continuity of the temperature, resulting in a substantially increased condition number of the system of linear equations, which deteriorates the efficiency of the method. Notice that, for very high-order accurate schemes this issue is critical and, therefore, the efficiency of the code is significantly deteriorated and, for that reason, such approach is not considered in the present work.

3.2.3 Polygonal meshes

A general polygonal mesh denoted as $\mathcal{M}^S$ discretizes each subdomain Ω^S and consists of n^S non-overlapping convex polygonal cells (triangles, quadrangles, etc.). Cells are denoted as c_i with $i \in \mathcal{I}^{\mathrm{A}} = \{1,\ldots,n^{\mathrm{A}}\}$ for cells belonging to mesh $\mathcal{M}^{\mathrm{A}}$ and $i \in \mathcal{I}^{\mathrm{B}} = \{n^{\mathrm{A}}+1,\ldots,n^{\mathrm{A}}+n^{\mathrm{B}}\}$ for cells belonging to mesh $\mathcal{M}^{\mathrm{B}}$ and, therefore, the index of each cell is unique.

Inner edges are denoted as e_{ij} with $j \neq i$ and $i,j \in \mathcal{I}^S$ and correspond to the edges shared between neighbor cells c_i and c_j belonging to mesh $\mathcal{M}^S$ and, therefore, $e_{ij} = c_i \cap c_j$. Boundary edges are denoted as e_{iF} with $i \in \mathcal{I}^{F,S}$ and correspond to the edges of cells c_i belonging to mesh $\mathcal{M}^S$ and approximating boundary subsets $\Gamma^{F,S}$ (for the sake of simplicity, each cell has at most one boundary edge). Subset $\mathcal{I}^{F,S} \subset \mathcal{I}^S$ gathers the indices and $n^{F,S}$ is the number of the cells belonging

to mesh $\mathcal{M}^S$ with a boundary edge approximating boundary subset $\Gamma^{F,S}$.

The vertices of the boundary edges fall on the curves of the associated boundary subsets and, in particular, on boundary subsets $\Gamma^{\mathrm{C},S}$ are common to both meshes (conformal meshes on the interface). In that case, for each boundary edge $e_{i\mathrm{C}}$ corresponding to an edge of cell c_i, with $i \in \mathcal{I}^{\mathrm{C,A}}$ and belonging to mesh $\mathcal{M}^{\mathrm{A}}$, there is a counterpart boundary edge $f_{j\mathrm{C}}$ corresponding to an edge of cell c_j, with $j \in \mathcal{I}^{\mathrm{C,B}}$ and belonging to mesh $\mathcal{M}^{\mathrm{B}}$, such that edges $f_{i\mathrm{C}}$ and $f_{j\mathrm{C}}$ share the same vertices and are geometrically equivalent (same shape and location).

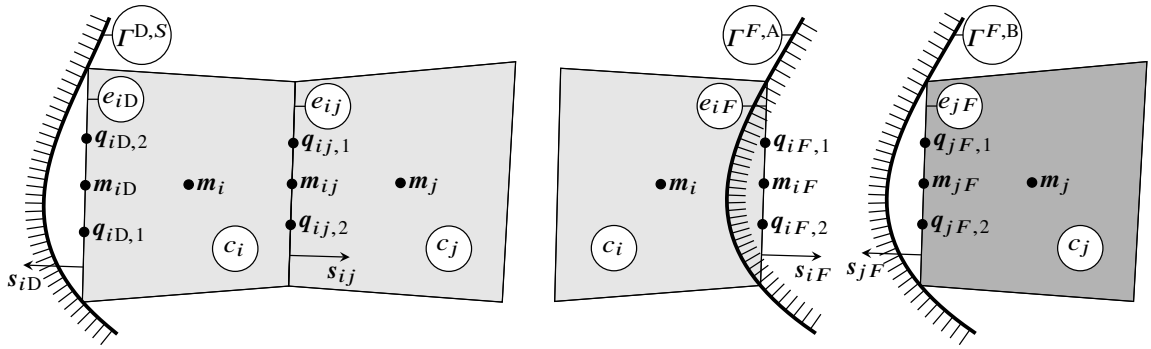
Table 3.1 introduces the geometric properties for the cells and edges and Figures 3.2 provides a schematic representation. Notice that inner edge e_{ij} is also denoted as e_{ji} and, therefore, reference and quadrature points are the same, that is, $\boldsymbol{m}_{ij} = \boldsymbol{m}_{ji}$ and $\boldsymbol{q}_{ij,r} = \boldsymbol{q}_{ji,r}$, whereas outward unit normal vectors are antisymmetric, that is, $\boldsymbol{s}_{ij} = -\boldsymbol{s}_{ji}$. Similarly, for counterpart boundary edges e_{iF} and e_{jF}, antisymmetric outward unit normal vectors are assigned, that is, $\boldsymbol{s}_{iF} = -\boldsymbol{s}_{jF}$.

Table 3.1: Notation and geometric properties for the cells and edges.

Mesh elements	Notation	Properties	Definition	Choice
Cells	$c_i \subset \Omega_\Delta^S$ $S \in \{\mathrm{A},\mathrm{B}\}$	∂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	$e_{ij} \in \Omega_\Delta^S$ $S \in \{\mathrm{A},\mathrm{B}\}$	$\lvert e_{ij} \rvert$	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	$e_{iF} \subset \Gamma_\Delta^{F,S}$ $F \in \{\mathrm{D},\mathrm{C},\mathrm{K}\}$ $S \in \{\mathrm{A},\mathrm{B}\}$	$\lvert e_{iF} \rvert$	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	

The computational subdomains correspond to the cells in meshes $\mathcal{M}^S$ discretizing physical subdomains Ω^S, denoted as Ω_Δ^S. For the computational subdomains, the computational boundaries correspond to the edges in meshes $\mathcal{M}^S$ discretizing physical boundaries Γ^S, denoted as Γ_Δ^S. For the computational boundaries, the computational boundary subsets correspond to the edges in meshes $\mathcal{M}^S$ discretizing physical boundary subsets $\Gamma^{F,S}$, denoted as $\Gamma_\Delta^{F,S}$. The computational subdomains, boundaries, and boundary subsets are, therefore, respectively given as

$$\Omega_\Delta^S = \bigcup_{i \in \mathcal{I}^S} c_i, \tag{3.7}$$



(a) Boundary. (b) Interface.

Figure 3.2: Notation and geometric properties for the cells and edges.

$$\Gamma_{\Delta}^{S} = \bigcup_{i \in \mathcal{I}^{F,S}, F \in \{\mathrm{D},\mathrm{C},\mathrm{K}\}} e_{iF}, \tag{3.8}$$

$$\Gamma_{\Delta}^{F,S} = \bigcup_{i \in \mathcal{I}^{F,S}} e_{iF}. \tag{3.9}$$

Since meshes $\mathcal{M}^S$ are composed of polygonal cells, computational subdomains Ω_{Δ}^{S} do not fully overlap associated curved physical subdomains Ω^S. For fine enough meshes, a geometrical mismatch with magnitude order $\mathcal{O}(h^2)$ is then expected between the physical and the computational boundaries, where h is the characteristic mesh size. Such geometrical mismatch leads to a potential accuracy deterioration and a maximal second-order of convergence for the method if the appropriate treatment is not provided.

3.3 Problem partitioning

The conjugate heat transfer problem is solved with a partitioning method applied to the problem model, which results in heat transfer subproblems governed by the convection-diffusion equation with only boundary conditions, denoted as subproblem P^S in subdomain Ω^S. The Dirichlet-Neumann and Neumann-Neumann methods are used as the partitioning methods, and Dirichlet and Neumann boundary conditions replace the prescribed interface conditions on each interface subset. Hence, the partitioned approach to solve the conjugate heat transfer problem produces a partitioned problem model that consists of Dirichlet and Neumann boundary value subproblems in the subdomains. Then, each subproblem is discretized separately with no explicit reference or direct dependency on the subproblem in the adjacent subdomain.

3.3.1 Partitioned problem model

The boundary of each subdomain comprises, not only boundary subsets $\Gamma^{\mathrm{D},S}$, but also interface subsets Γ^{C} and Γ^{K} of domain Ω (the interface of the entire domain is part of the boundaries of the

subdomains). For notation convenience, $\Gamma^{F,S}$, $F \in \{\mathrm{D},\mathrm{C},\mathrm{K}\}$, denotes the subsets of the partitioned boundaries of subdomains Ω^S such that $\Gamma^S = \Gamma^{\mathrm{D},S} \cup \Gamma^{\mathrm{C},S} \cup \Gamma^{\mathrm{K},S}$ (see Figure 3.3). Notice that boundary subsets $\Gamma^{\mathrm{C},S}$ and $\Gamma^{\mathrm{K},S}$ of subdomains Ω^S correspond to interface subsets Γ^{C} and Γ^{K} of domain Ω, that is, $\Gamma^{\mathrm{C,A}} = \Gamma^{\mathrm{C,B}} = \Gamma^{\mathrm{C}}$ and $\Gamma^{\mathrm{K,A}} = \Gamma^{\mathrm{K,B}} = \Gamma^{\mathrm{K}}$. From now on, the term boundaries refer to subdomain boundaries Γ^S rather than to only domain boundary Γ^{D}, unless otherwise stated.

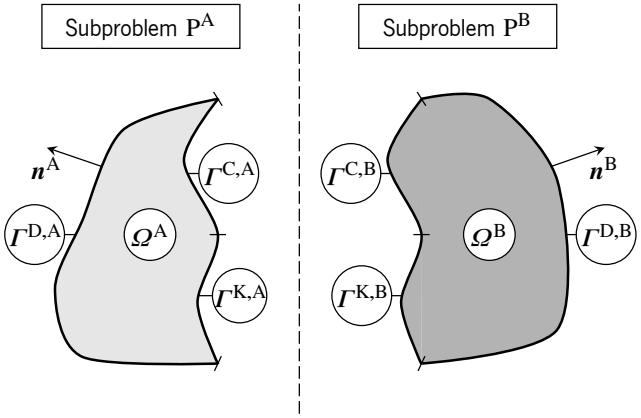


Figure 3.3: Example of subproblems in arbitrary curved subdomains with boundary subsets and outward unit normal vectors.

In the case of the continuity interface conditions, the Dirichlet-Neumann method results in a Dirichlet boundary condition prescribed in subproblem P^{A} and a Neumann boundary condition prescribed in subproblem P^{B}. Similarly, in the case of the imperfect interface conditions, the Neumann-Neumann method results in Neumann boundary conditions prescribed in both subproblems, P^{A} and P^{B}.

Subproblem P^{A} consists in seeking temperature function $\phi^{\mathrm{A}}(\boldsymbol{x})$ in subdomain Ω^{A} with governing partial differential equation and boundary conditions given as

$$\mathrm{P}^{\mathrm{A}}: \begin{cases} \nabla \cdot \left(\boldsymbol{u}^{\mathrm{A}} \phi^{\mathrm{A}} - \kappa^{\mathrm{A}} \nabla \phi^{\mathrm{A}} \right) = f^{\mathrm{A}}, & \text{in } \Omega^{\mathrm{A}}, \quad (3.10) \\ \phi^{\mathrm{A}} = g^{\mathrm{D,A}}, & \text{on } \Gamma^{\mathrm{D,A}}, \quad (3.11) \\ \phi^{\mathrm{A}} = g^{\mathrm{C,A}}, & \text{on } \Gamma^{\mathrm{C,A}}, \quad (3.12) \\ -\kappa^{\mathrm{A}} \nabla \phi^{\mathrm{A}} \cdot \boldsymbol{n}^{\mathrm{A}} = g^{\mathrm{K,A}}, & \text{on } \Gamma^{\mathrm{K,A}}, \quad (3.13) \end{cases}$$

where $g^{\mathrm{C,A}} := g^{\mathrm{C,A}}(\boldsymbol{x})$ and $g^{\mathrm{K,A}} := g^{\mathrm{K,A}}(\boldsymbol{x})$ are unknown regular and bounded functions on boundary subsets $\Gamma^{\mathrm{C,A}}$ and $\Gamma^{\mathrm{K,A}}$, respectively. On the other side, subproblem P^{B} consists in seeking temperature function $\phi^{\mathrm{B}}(\boldsymbol{x})$ in subdomain Ω^{B} with governing partial differential equation and

boundary conditions given as

$$\mathrm{P}^{\mathrm{B}}: \begin{cases} \nabla \cdot \left(\boldsymbol{u}^{\mathrm{B}} \phi^{\mathrm{B}} - \kappa^{\mathrm{B}} \nabla \phi^{\mathrm{B}} \right) = f^{\mathrm{B}}, & \text{in } \Omega^{\mathrm{B}}, \quad (3.14) \\ \phi^{\mathrm{B}} = g^{\mathrm{D,B}}, & \text{on } \Gamma^{\mathrm{D,B}}, \quad (3.15) \\ -\kappa^{\mathrm{B}} \nabla \phi^{\mathrm{B}} \cdot \boldsymbol{n}^{\mathrm{B}} = g^{\mathrm{C,B}}, & \text{on } \Gamma^{\mathrm{C,B}}, \quad (3.16) \\ -\kappa^{\mathrm{B}} \nabla \phi^{\mathrm{B}} \cdot \boldsymbol{n}^{\mathrm{B}} = g^{\mathrm{K,B}}, & \text{on } \Gamma^{\mathrm{K,B}}, \quad (3.17) \end{cases}$$

where $g^{\mathrm{C,B}} := g^{\mathrm{C,B}}(\boldsymbol{x})$ and $g^{\mathrm{K,B}} := g^{\mathrm{K,B}}(\boldsymbol{x})$ are unknown regular and bounded functions on boundary subsets $\Gamma^{\mathrm{C,B}}$ and $\Gamma^{\mathrm{K,B}}$, respectively. The condition $g^{\mathrm{K,A}}(\boldsymbol{x}) + g^{\mathrm{K,B}}(\boldsymbol{x}) = 0$ holds on interface subset Γ^{K}.

Notice that, Dirichlet boundary condition (3.2) prescribed on boundary subsets $\Gamma^{\mathrm{D},S}$ correspond to Dirichlet boundary conditions (3.11) and (3.15) in subproblems P^{A} and P^{B}, respectively. The continuity interface conditions (3.3) and (3.4) prescribed on interface subset Γ^{C} are replaced with Dirichlet boundary condition (3.12) in subproblem P^{A} and Neumann boundary condition (3.16) in subproblem P^{B}, and the associated boundary condition values are given with unknown functions $g^{\mathrm{C,A}}(\boldsymbol{x})$ and $g^{\mathrm{C,B}}(\boldsymbol{x})$, respectively. The imperfect interface conditions (3.5) and (3.6) prescribed on interface subset Γ^{K} are replaced with Neumann boundary conditions (3.13) and (3.17) in subproblems P^{A} and P^{B}, respectively, and the associated boundary condition values are given with unknown functions $g^{\mathrm{K,A}}(\boldsymbol{x})$ and $g^{\mathrm{K,B}}(\boldsymbol{x})$, respectively.

3.3.2 Thermal coupling

The partitioning of the conjugate heat transfer problem with the Dirichlet-Neumann and Neumann-Neumann methods results in subproblems with only boundary conditions prescribed on the boundary subsets of the associated subdomains. However, the produced Dirichlet and Neumann boundary conditions in Equations (3.12) and (3.13) and (3.16) and (3.17), resulting from the continuity and imperfect interface conditions, brings to the problem additional unknown functions $g^{F,S}(\boldsymbol{x})$, $F \in \{\mathrm{C}, \mathrm{K}\}$, to provide the boundary condition values on the associated boundary subsets.

The strategy to guarantee the appropriate thermal coupling between the subproblems is to provide boundary condition functions $g^{F,S}(\boldsymbol{x})$ such that the solution of the partitioned problem satisfies the interface conditions and, therefore, corresponds to the solution of the conjugate heat transfer problem. Therefore, Equations (3.12) and (3.16) recover the continuity interface conditions (3.3) and (3.4) while Equations (3.13) and (3.17) recover the imperfect interface conditions (3.5) and (3.6). This is achieved assigning boundary condition functions $g^{F,S}(\boldsymbol{x})$ given as

$$g^{\mathrm{C,A}} := \phi^{\mathrm{B}}, \quad \text{on } \Gamma^{\mathrm{C,A}}, \quad (3.18)$$

$$g^{\mathrm{C,B}} := -\kappa^{\mathrm{A}} \nabla \phi^{\mathrm{A}} \cdot \boldsymbol{n}^{\mathrm{B}}, \quad \text{on } \Gamma^{\mathrm{C,B}}, \quad (3.19)$$

$$g^{\mathrm{K,A}} := h^{\mathrm{K}}\left(\phi^{\mathrm{A}} - \phi^{\mathrm{B}}\right), \quad \text{on } \Gamma^{\mathrm{K,A}}, \tag{3.20}$$

$$g^{\mathrm{K,B}} := h^{\mathrm{K}}\left(\phi^{\mathrm{B}} - \phi^{\mathrm{A}}\right), \quad \text{on } \Gamma^{\mathrm{K,B}}, \tag{3.21}$$

where the symmetry $g^{\mathrm{K,A}}(\boldsymbol{x}) = -g^{\mathrm{K,B}}(\boldsymbol{x})$ is noticed.

Although it seems that the partitioned problem model is transformed into the conjugate model problem, which is partitioned in the first place, notice that functions $g^{F,S}(\boldsymbol{x})$ are unknowns of the former and do not exist in the latter. Indeed, these functions serve as transmission conditions on the interface and, therefore, subproblems are discretized separately with no explicit reference or direct dependence on the subproblem in the adjacent subdomain. In turn, an additional numerical treatment is required to determine unknown boundary condition functions $g^{F,S}(\boldsymbol{x})$ since temperature functions $\phi^{S}(\boldsymbol{x})$ are also unknown.

Remark 11 Boundary condition function $g^{C,A}(\boldsymbol{x})$ requires the solution of subproblem P^{B}, boundary condition function $g^{C,B}(\boldsymbol{x})$ requires the solution of subproblem P^{A}, and boundary condition functions $g^{K,S}(\boldsymbol{x})$ require the solution of both subproblems, which seems to imply that one subproblem can only be solved after solution of the other. Indeed, a naive strategy consists in using an iterative algorithm to solve the subproblems consecutively [8], until the solution of both have converged, assigning boundary condition functions $g^{F,S}(\boldsymbol{x})$ based on the solution of the subproblems in the previous iteration. On the other side, since the partitioned model problem is linear, boundary condition functions $g^{F,S}(\boldsymbol{x})$ can be determined in the same system together with temperature functions $\phi^{S}(\boldsymbol{x})$, which avoids the drawbacks of iterative algorithms.

Remark 12 The Dirichlet-Neumann method applied for the continuity interface conditions yields two asymmetric subproblems with different types of boundary conditions on boundary subsets $\Gamma^{C,S}$ of subdomains Ω^{S}. Namely, a Dirichlet boundary condition is prescribed on boundary subset $\Gamma^{C,A}$ in subproblem P^{A} and a Neumann boundary condition is prescribed on boundary subset $\Gamma^{C,B}$ in subproblem P^{B}. However, Neumann boundary condition can be prescribed in the former, and a Dirichlet boundary condition can be prescribed in latter, which is equivalent in terms of the thermal coupling for the partitioned problem. On the other side, the boundary conditions configuration can be an issue in terms of accuracy and robustness of the discretized subproblems. Therefore, although the first configuration is selected to present the proposed method, both are addressed in the numerical benchmark provided in this work to investigate the accuracy and robustness implications depending on the thermal conductivity assigned for the adjacent subdomains.

3.4 Polynomial reconstructions

The treatment of partitioned problem consists in computing polynomial reconstructions to provide local approximations of the temperature functions in the associated computational subdomains.

Additionally, they also provide approximations of the boundary condition functions resulting from the partitioning method on the associated computational boundary subsets. Unconstrained polynomial reconstructions are computed associated with each inner edge, whereas constrained polynomial reconstructions for cell mean-value conservation are computed associated with each cell. Additionally, constrained polynomial reconstructions for the boundary conditions fulfilment are computed associated with each boundary edge and, in the case of arbitrary curved boundaries, the ROD method is applied to preserve the accuracy of the polynomial reconstructions. These unconstrained and constrained polynomial reconstructions and the application of the ROD method are detailed hereafter.

3.4.1 Polynomial reconstruction method

Temperature functions are represented in terms of approximate and piecewise cell mean-values in the associated computational subdomain. That is, for cell c_i in computational subdomain Ω_Δ^S, associated cell mean-value, denoted as ϕ_i, is defined as

$$\phi_i \approx \frac{1}{|c_i|}\int_{c_i} \phi^S(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{3.22}$$

and vector $\boldsymbol{\Phi}^S \in \mathbb{R}^{n^S}$ gathers the cell mean-values in computational subdomain Ω_Δ^S. Notice that, although the numerical approximations are given in computational subdomains Ω_Δ^S, the associated temperature functions are defined in physical subdomains Ω^S, which do not entirely overlap for curved domains.

The polynomial reconstruction method consists in fitting through a minimization procedure a polynomial function of degree d to a certain number of cell mean-values of the temperature function [67–69]. For regular functions, the polynomial reconstruction method provides local approximations with an error with magnitude order $\mathcal{O}\left(h^{d+1}\right)$, where h is the mesh characteristic size, therefore, convergence order $d+1$ is expected under mesh refinement [63–66, 70–72]. A generalization of the method is introduced in Costa et al., 2018 [65], and enables to compute local approximations that, besides approximating the cell mean-values, are also capable of fulfilling specific conditions associated with the problem.

Consider a polynomial function of degree d denoted as $\varphi(\boldsymbol{x})$ and written in a compacted form as

$$\varphi(\boldsymbol{x}) = \boldsymbol{\eta}\cdot\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}) = \sum_{\alpha=0}^{d}\sum_{\beta=0}^{d-\alpha}\eta_{\alpha\beta}(x-m_x)^\alpha\left(y-m_y\right)^\beta, \tag{3.23}$$

where $\boldsymbol{p}_d(\boldsymbol{x})$ is a basis function vector including all two-dimensional monomials up to degree d, $\boldsymbol{m} = \left(m_x, m_y\right)$ is a reference point in order to reduce condition numbers, and vector $\boldsymbol{\eta} \in \mathbb{R}^n$ gathers the polynomial coefficients, with $n = (d+1)(d+2)/2$.

The polynomial reconstruction method requires seeking, through a minimization procedure, polynomial coefficients vector $\boldsymbol{\eta}$ that provides the best approximation of polynomial function $\varphi(\boldsymbol{x})$ to the cell mean-values in the vicinity of the reference point and the same subdomain. The method provides either unconstrained or constrained polynomial reconstructions depending on the minimization procedure performed to compute polynomial coefficients vector $\boldsymbol{\eta}$. An unconstrained polynomial reconstruction corresponds to the best approximation to the cell mean-values. On the other side, a constrained polynomial reconstruction corresponds to the best approximation to the cell mean-values given that specific conditions are fulfilled (called constraints).

For each subproblem, unconstrained and constrained polynomial reconstructions are computed, which allow obtaining local continuous and differentiable functions that approximate the associated temperature functions. The polynomial reconstructions for each subproblem are only computed with the cell mean-values associated with the same subproblem.

3.4.2 Least-squares method

The linear least-squares method is the standard minimization procedure performed to provide the polynomial reconstructions. The method consists in minimizing the sum of the squared residuals, where the residuals are the difference between the cell mean-values of the polynomial function and the cell mean-values of the associated temperature function. For that purpose, consider weighted cost functional from $\mathbb{R}^n$ to $\mathbb{R}$, denoted as $F(\boldsymbol{\eta})$, given as

$$F(\boldsymbol{\eta}) = \sum_{k \in \mathcal{S}} \omega_k \left[\frac{1}{|c_k|} \int_{c_k} \varphi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \phi_k \right]^2, \tag{3.24}$$

where, for the sake of simplicity without loss of accuracy, the integrals are evaluated numerically using quadrature rules of the same order as the polynomial degree.

In weighted cost functional (3.24), $\mathcal{S}$ is a stencil gathering the indices of s cells in the vicinity of the reference point such that the associated mean-values are considered in the approximation. For polynomial degree d, there are n polynomial coefficients to determine and, therefore, $s \geq n$ is required such that the number of mean-values to approximate is, at least, the same as the number of polynomial coefficients to determine. The case of $s = n$ corresponds to the exact fitting of the polynomial function to the mean-values. In contrast, the common practice consists in considering $s = 1.5n$ to exceed sufficiently the number of polynomial coefficients, which provides robustness and stability to the polynomial reconstruction in the general case.

Moreover, in weighted cost functional (3.24), ω_k, $k \in \mathcal{S}$, are positive weights aiming to give a relative importance to the mean-values according to the Euclidean distance between associated cell reference point $\boldsymbol{m}_k$ and polynomial reconstruction reference point $\boldsymbol{m}$, given as $d_k = |\boldsymbol{m}_k - \boldsymbol{m}|$, as applied in Barth, 1993 [68]. The weights are defined as $\omega_k := \omega(d_k)$, $k \in \mathcal{S}$, with function $\omega(d_k)$

given as

$$\omega(d_k) = \frac{1}{(\sigma d_k)^{\delta} + 1}, \tag{3.25}$$

where $\delta, \sigma \in \mathbb{R}$ are given parameters and vector $\boldsymbol{\omega} \in \mathbb{R}^s$ gathers the weights associated with the cells with an index in the stencil.

Polynomial reconstructions for the subproblems are computed based on stencils that gather indices of cells solely in the associated computational subdomain. That is, a polynomial reconstruction for subproblem P^S approximates temperature function $\phi(\boldsymbol{x})$ and, therefore, only mean-values associated with cells in computational subdomain Ω^S are considered for the minimization procedure. Therefore, each index k in stencil $\mathcal{S}$ corresponds to cell c_k in computational subdomain Ω^S_Δ and, therefore, $\mathcal{S} \subset \mathcal{I}^S$. This procedure is necessary to avoid approximating functions or derivatives with discontinuities on the interface, which are not correctly approximated with polynomial reconstructions without deterioration of the convergence order of the method. Moreover, the consistency of the polynomial reconstruction method is maintained since functions represented with polynomial terms are exactly fitted with polynomial reconstructions with the same degree.

For unconstrained polynomial reconstructions, the minimization procedure consists in seeking unique vector $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes weighted cost functional $F(\boldsymbol{\eta})$ in the least-squares sense, that is, $\widetilde{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} F(\boldsymbol{\eta})$, and the optimal polynomial reconstruction is written as $\widetilde{\varphi}(\boldsymbol{x}) = \widetilde{\boldsymbol{\eta}} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m})$. The normal equations method [73] is used to derive the unconstrained least-squares method and perform the associated minimization procedure in the present work.

For constrained polynomial reconstructions, in addition to minimizing functional $F(\boldsymbol{\eta})$ in the least-squares sense, a linear constraint derived from specific conditions associated with the problem is exactly fulfilled. Consider a linear constraint functional from $\mathbb{R}^n$ to $\mathbb{R}$, denoted as $G(\boldsymbol{\eta})$, gathering the linear constraints in the residual form. The minimization procedure consists in seeking unique vector $\widehat{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes weighted cost functional $F(\boldsymbol{\eta})$ in the least-squares sense and exactly fulfills equation $G(\boldsymbol{\eta}) = 0$, that is, $\widehat{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} F(\boldsymbol{\eta})$ subject to $G(\boldsymbol{\eta}) = 0$, and the optimal polynomial reconstruction is written as $\widehat{\varphi}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m})$. The so-called linearly constrained Lagrange multipliers method [74] is used to derive the constrained least-squares method and perform the associated minimization procedure in the present work.

3.4.3 Unconstrained polynomial reconstructions

For each inner edge e_{ij} in computational subdomain Ω^S_Δ, an unconstrained polynomial reconstruction is assigned and denoted as $\varphi_{ij}(\boldsymbol{x}) = \boldsymbol{\eta}_{ij} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{ij})$. Vector $\boldsymbol{\eta} \coloneqq \boldsymbol{\eta}_{ij}$ gathers the polynomial coefficients and reference point $\boldsymbol{m} \coloneqq \boldsymbol{m}_{ij}$ corresponds to the edge midpoint. For the minimization procedure, weighted cost functional $F(\boldsymbol{\eta})$ is denoted as $F_{ij}(\boldsymbol{\eta}_{ij})$ where stencil $\mathcal{S} \coloneqq \mathcal{S}_{ij}$ gathers the indices of s cells in the vicinity of the reference point (belonging to computational

subdomain Ω_{Δ}^{S}) and vector $\boldsymbol{\omega} \coloneqq \boldsymbol{\omega}_{ij}$ gathers the weights associated with each cell in the stencil.

Following the polynomial reconstruction method, vector $\widetilde{\boldsymbol{\eta}}_{ij} \in \mathbb{R}^n$ minimizes weighted cost functional $F\left(\boldsymbol{\eta}_{ij}\right)$, and the optimal unconstrained polynomial reconstruction is written as $\widetilde{\varphi}_{ij}(\boldsymbol{x}) = \widetilde{\boldsymbol{\eta}}_{ij} \cdot \boldsymbol{p}_d\left(\boldsymbol{x} - \boldsymbol{m}_{ij}\right)$.

3.4.4 Constrained polynomial reconstructions for cells

For each cell c_i in computational subdomain Ω_{Δ}^{S}, a constrained polynomial reconstruction is assigned and denoted as $\varphi_i(\boldsymbol{x}) = \boldsymbol{\eta}_i \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_i)$. Vector $\boldsymbol{\eta} \coloneqq \boldsymbol{\eta}_i$ gathers the polynomial coefficients and reference point $\boldsymbol{m} \coloneqq \boldsymbol{m}_i$ corresponds to the cell centroid. For the minimization procedure, weighted cost functional $F(\boldsymbol{\eta})$ is denoted as $F_i\left(\boldsymbol{\eta}_i\right)$ where stencil $\mathcal{S} \coloneqq \mathcal{S}_i$ gathers the indices of s cells in the vicinity of the reference point (belonging to computational subdomain Ω_{Δ}^{S}) and vector $\boldsymbol{\omega} \coloneqq \boldsymbol{\omega}_i$ gathers the weights associated with each cell in the stencil.

The polynomial reconstruction is constrained such that it recovers the cell mean-value of temperature function $\phi^S(\boldsymbol{x})$. That is, the integral of the polynomial reconstruction in cell c_i (divided by the area of the same cell) equals cell mean-value ϕ_i and, therefore, constraint functional $G(\boldsymbol{\eta}) \coloneqq G_i\left(\boldsymbol{\eta}_i\right)$ is given as

$$G_i\left(\boldsymbol{\eta}_i\right) = \frac{1}{|c_i|}\int_{c_i} \varphi_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \phi_i. \tag{3.26}$$

Following the polynomial reconstruction method, vector $\widetilde{\boldsymbol{\eta}}_i \in \mathbb{R}^n$ minimizes weighted cost functional $F\left(\boldsymbol{\eta}_i\right)$ subject to equation $G_i\left(\boldsymbol{\eta}_i\right) = 0$, and the optimal constrained polynomial reconstruction is written as $\widehat{\varphi}_i(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_i \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_i)$.

3.4.5 Constrained polynomial reconstructions for boundary edges

For each boundary edge e_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$ of computational subdomain Ω_{Δ}^{S}, a constrained polynomial reconstruction is assigned and denoted as $\varphi_{iF}(\boldsymbol{x}) = \boldsymbol{\eta}_{iF} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{iF})$. Vector $\boldsymbol{\eta} \coloneqq \boldsymbol{\eta}_{iF}$ gathers the polynomial coefficients and reference point $\boldsymbol{m} \coloneqq \boldsymbol{m}_{iF}$ corresponds to the edge midpoint. For the minimization procedure, weighted cost functional $F(\boldsymbol{\eta})$ is denoted as $F_{iF}\left(\boldsymbol{\eta}_{iF}\right)$ where stencil $\mathcal{S} \coloneqq \mathcal{S}_{iF}$ gathers the indices of s cells in the vicinity of the reference point (belonging to computational subdomain Ω_{Δ}^{S}) and vector $\boldsymbol{\omega} \coloneqq \boldsymbol{\omega}_{iF}$ gathers the weights associated with each cell in the stencil.

Moreover, $\boldsymbol{p}_{iF} = \left(p_{iF,x}, p_{iF,y}\right)$ is a collocation point in the vicinity of the reference point, $\boldsymbol{v}_{iF} = \left(v_{iF,x}, v_{iF,y}\right)$ is a unit vector, and g_{iF} is the point-value of the prescribed boundary condition function. The polynomial reconstruction is constrained such that it recovers the point-value of the prescribed boundary condition function at collocation point $\boldsymbol{p}_{iF}$. That is, the determination of the prescribed boundary condition with polynomial reconstruction at collocation point $\boldsymbol{p}_{iF}$ equals

point-value g_{iF} and, therefore, prototype constraint functional $G(\boldsymbol{\eta}) \coloneqq G_{iF}(\boldsymbol{\eta}_{iF})$ is given as

$$G_{iF}(\boldsymbol{\eta}_{iF}) = \varphi_{iF}(\boldsymbol{p}_{iF}) - g_{iF}, \tag{3.27}$$

in the case of a prescribed Dirichlet boundary condition and is given as

$$G_{iF}(\boldsymbol{\eta}_{iF}) = -\kappa^{S}(\boldsymbol{p}_{iF})\nabla\varphi_{iF}(\boldsymbol{p}_{iF}) \cdot \boldsymbol{v}_{iF} - g_{iF}, \tag{3.28}$$

in the case of a prescribed Neumann boundary condition.

Following the polynomial reconstruction method, vector $\widetilde{\boldsymbol{\eta}}_{iF} \in \mathbb{R}^n$ minimizes weighted cost functional $F(\boldsymbol{\eta}_{iF})$ subject to equation $G_{iF}(\boldsymbol{\eta}_{iF}) = 0$, and the optimal constrained polynomial reconstruction is written as $\widehat{\varphi}_{iF}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_{iF} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{iF})$.

Notice that the same constrained least-squares method is always used to perform the minimization procedure and compute the constrained polynomial reconstructions associated with the boundary edges. In contrast, only the prototype constraint functional is given following the prescribed boundary condition type.

3.4.6 Reconstruction for off-site data method

The ROD method provides constrained polynomial reconstructions associated with the boundary edges such that the prescribed boundary conditions are exactly fulfilled on the associated curved physical boundaries. Therefore, the ROD method is capable of handling the geometrical mismatch between the straight computational and curved physical boundaries to preserve the convergence order and, thus, avoids the difficulties associated with the approaches with curved meshes. Moreover, the ROD method does not depend on the prescribed boundary condition type, and the full parameterization of the boundary is not required, which makes this method simple, versatile and general for the treatment of curved physical boundaries.

The ROD method consists in providing the appropriate linear constraints to the polynomial reconstructions associated with the boundary edges assigning collocation points and unit vectors on the curved physical boundaries. Therefore, the applied constrained least-squares method to perform the minimization procedure provides polynomial reconstructions that exactly fulfil the associated boundary conditions on the curved physical boundaries.

Polynomial reconstruction $\widehat{\varphi}_{iF}(\boldsymbol{x})$ associated with boundary edge e_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$ requires collocation point $\boldsymbol{p}_{iF}$ and unit vector $\boldsymbol{v}_{iF}$ for the constraint functional given in Equation (3.27) or (3.28). To properly define the linear constraint for the minimization procedure, the ROD method consists in assigning collocation point $\boldsymbol{p}_{iF} \coloneqq \boldsymbol{b}_{iF}$ and unit vector $\boldsymbol{v}_{iF} \coloneqq \boldsymbol{n}_{iF} = \boldsymbol{n}^{S}(\boldsymbol{b}_{iF})$ belonging to curved physical boundary subset $\Gamma^{F,S}$ (see Figure 3.4). Moreover, point-values g_{iF} of the prescribed boundary condition function in Equations (3.27) and (3.28) are also defined at

collocation point $\boldsymbol{b}_{iF}$. In the present study, collocation point $\boldsymbol{b}_{iF}$ is obtained from the projection of edge midpoint $\boldsymbol{m}_{iF}$ onto the associated physical boundary subset and vector $\boldsymbol{n}_{iF}$ corresponds to the unit vector normal to the same physical boundary subset at point $\boldsymbol{b}_{iF}$. However, different choices can be considered, whereas the only limitation is that the collocation point and unit vector are obtained from the curved physical boundary. Notice that, for counterpart boundary edges e_{iF} and e_{iF} discretizing physical boundary subsets $\Gamma^{F,\mathrm{A}}$ and $\Gamma^{F,\mathrm{B}}$, $F \in \{\mathrm{C},\mathrm{K}\}$, respectively, collocation points coincide and outward unit vectors are symmetric, that is, $\boldsymbol{b}_{iF} = \boldsymbol{b}_{jF}$ and $\boldsymbol{n}_{iF} = -\boldsymbol{n}_{jF}$.

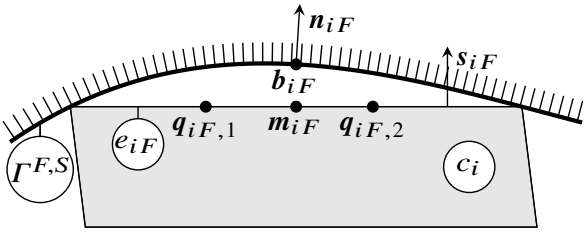


Figure 3.4: Representation of a curved physical boundary fitted with a polygonal cell.

Remark 13 The common practice to fulfil the prescribed boundary conditions without a specific treatment of curved physical boundaries is to assign collocation points and unit vectors from the computational boundary subset. For instance, the collocation points correspond to edge midpoints and the unit vectors correspond to edge normal vectors, that is, $\boldsymbol{p}_{iF} \coloneqq \boldsymbol{m}_{iF}$ and $\boldsymbol{v}_{iF} \coloneqq \boldsymbol{s}_{iF}$. Therefore, such practice provides a maximal second-order of convergence in the case of curved physical boundaries since the geometrical mismatch between the computational and the physical boundaries has magnitude order $\mathcal{O}(h^2)$, with h the characteristic mesh size. However, the optimal convergence order is preserved when the computational and physical boundaries exactly match, that is, only in the case polygonal domains.

3.4.6.1 Boundary edges on Dirichlet computational boundary subset

For boundary edge $e_{i\mathrm{D}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{D},S}$, associated polynomial reconstruction $\varphi_{i\mathrm{D}}(\boldsymbol{x})$ exactly fulfills Dirichlet boundary condition (3.11) or (3.15) in subproblems P^{A} and P^{B}, respectively. Therefore, prototype constraint functional (3.27) is considered for the minimization procedure with collocation point $\boldsymbol{p}_{iF} \coloneqq \boldsymbol{b}_{i\mathrm{D}}$ and point-value $g_{iF} \coloneqq g_{i\mathrm{D}} = g^{\mathrm{D},S}(\boldsymbol{b}_{i\mathrm{D}})$ of boundary condition function $g^{\mathrm{D},S}(\boldsymbol{x})$ (see Figure 3.5).

3.4.6.2 Boundary edges on continuity computational boundary subset

For boundary edge $e_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C},\mathrm{A}}$, associated polynomial reconstruction $\varphi_{i\mathrm{C}}(\boldsymbol{x})$ exactly fulfills Dirichlet boundary condition (3.12) in subproblem P^{A}. Therefore, prototype constraint functional (3.27) is considered for the minimization procedure with collocation

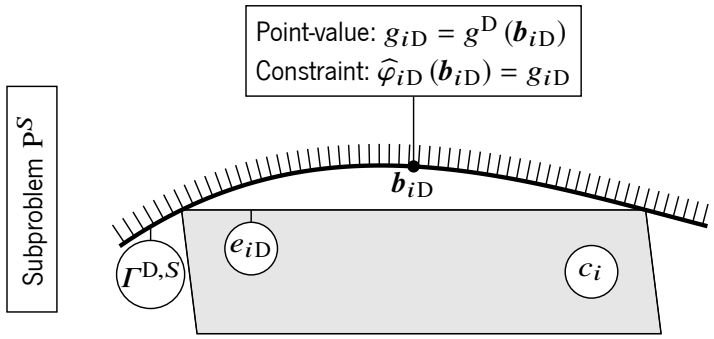


Figure 3.5: Representation of the collocation point and the least-squares constraint for the ROD method on Dirichlet computational boundary subsets.

point $\boldsymbol{b}_{iF} \coloneqq \boldsymbol{b}_{i\mathrm{C}}$ and point-value $g_{iF} \coloneqq g_{i\mathrm{C}} \approx g^{\mathrm{C,A}}(\boldsymbol{b}_{i\mathrm{C}})$ approximating boundary condition function $g^{\mathrm{C,A}}(\boldsymbol{x})$ (see Figure 3.6).

For boundary edge $e_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$, associated polynomial reconstruction $\varphi_{j\mathrm{C}}(\boldsymbol{x})$ exactly fulfills Neumann boundary condition (3.16) in subproblem P^{B}. Therefore, prototype constraint functional (3.28) is considered for the minimization procedure with collocation point $\boldsymbol{b}_{jF} \coloneqq \boldsymbol{b}_{j\mathrm{C}}$, unit normal vector $\boldsymbol{n}_{jF} \coloneqq \boldsymbol{n}_{j\mathrm{C}}$, and point-value $g_{jF} \coloneqq g_{j\mathrm{C}} \approx g^{\mathrm{C,B}}(\boldsymbol{b}_{j\mathrm{C}})$ approximating boundary condition function $g^{\mathrm{C,B}}(\boldsymbol{x})$ (see Figure 3.6).

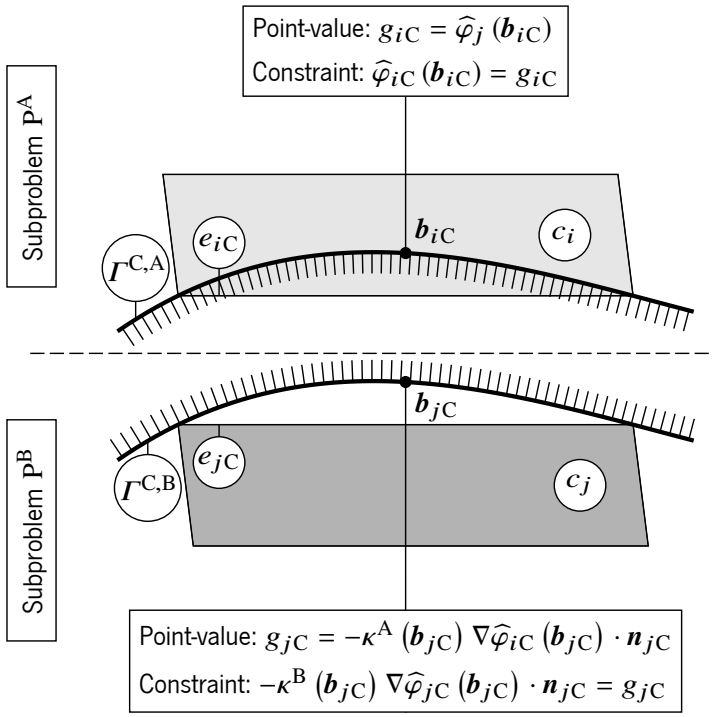


Figure 3.6: Representation of the collocation point and the least-squares constraint for the ROD method on continuity computational boundary subset.

3.4.6.3 Boundary edges on Kapitza computational boundary subset

For boundary edge $e_{i\mathrm{K}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{K,A}}$, associated polynomial reconstruction $\varphi_{i\mathrm{K}}(\boldsymbol{x})$ exactly fulfills Neumann boundary condition (3.13) in subproblem P^{A}. Therefore, prototype constraint functional (3.28) is considered for the minimization procedure with collocation point $\boldsymbol{b}_{iF} \coloneqq \boldsymbol{b}_{i\mathrm{K}}$, unit normal vector $\boldsymbol{n}_{iF} \coloneqq \boldsymbol{n}_{i\mathrm{K}}$, and point-value $g_{iF} \coloneqq g_{i\mathrm{K}} \approx g^{\mathrm{K,A}}(\boldsymbol{b}_{i\mathrm{K}})$ approximating boundary condition function $g^{\mathrm{K,A}}(\boldsymbol{x})$ (see Figure 3.7).

For boundary edge $e_{j\mathrm{K}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{K,B}}$, associated polynomial reconstruction $\varphi_{i\mathrm{K}}(\boldsymbol{x})$ exactly fulfills Neumann condition (3.17) in subproblem P^{B}. Therefore, prototype constraint functional (3.28) is considered for the minimization procedure with collocation point $\boldsymbol{b}_{jF} \coloneqq \boldsymbol{b}_{j\mathrm{K}}$, unit normal vector $\boldsymbol{n}_{jF} \coloneqq \boldsymbol{n}_{j\mathrm{K}}$, and point-value $g_{jF} \coloneqq g_{j\mathrm{K}} \approx g^{\mathrm{K,B}}(\boldsymbol{b}_{j\mathrm{K}})$ approximating boundary condition function $g^{\mathrm{K,B}}(\boldsymbol{x})$ (see Figure 3.7).

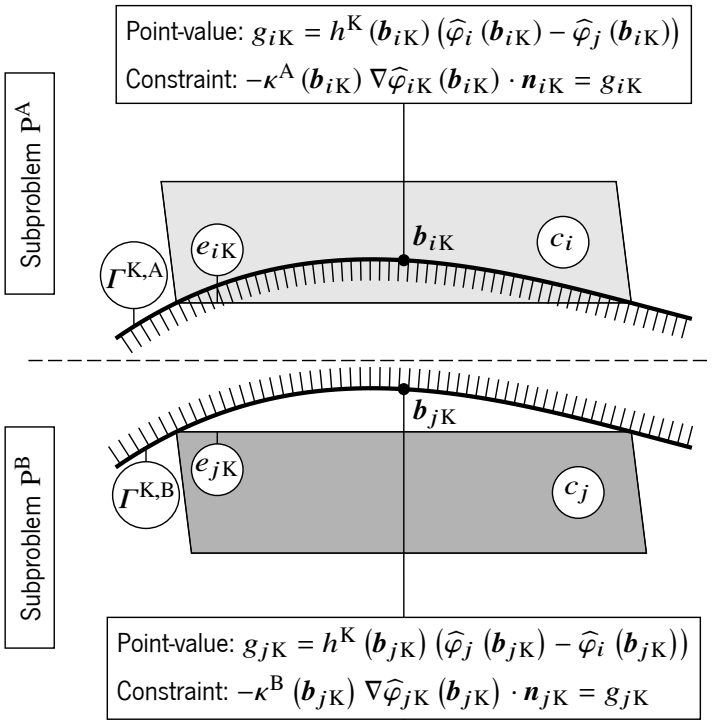


Figure 3.7: Representation of the collocation point and the least-squares constraint for the ROD method on Kapitza computational boundary subset.

Table 3.2 summarizes the required polynomial reconstructions for the partitioned problem treatment associated with the cells, inner edges, and boundary edges and the associated notation, subproblem, and constraint functional.

3.5 Finite volume discretization

A finite volume method is used to discretize the Dirichlet and Neumann boundary value subproblems in curved physical domains with the use of general polygonal meshes. The very

Table 3.2: Polynomial reconstructions associated with the cells, inner edges, and boundary edges.

Mesh elements	Subset	Notation	Subproblem	Constraint functional
Inner edges	$e_{ij} \subset \Omega_{\Delta}^{S}$ $S \in \{\mathrm{A},\mathrm{B}\}$	$\widetilde{\varphi}_{ij}(\boldsymbol{x})$	P^{S}	None (unconstrained)
Cells	$c_i \subset \Omega_{\Delta}^{S}$ $S \in \{\mathrm{A},\mathrm{B}\}$	$\widehat{\varphi}_{i}(\boldsymbol{x})$	P^{S}	Cell mean-value conservation, Equation (3.26)
Boundary edges	$e_{i\mathrm{D}} \subset \Gamma_{\Delta}^{\mathrm{D},S}$ $S \in \{\mathrm{A},\mathrm{B}\}$	$\widehat{\varphi}_{i\mathrm{D}}(\boldsymbol{x})$	P^{S}	Dirichlet boundary condition, Equation (3.27)
	$e_{i\mathrm{C}} \subset \Gamma_{\Delta}^{\mathrm{C,A}}$	$\widehat{\varphi}_{i\mathrm{C}}(\boldsymbol{x})$	P^{A}	Dirichlet boundary condition, Equation (3.27)
	$e_{j\mathrm{C}} \subset \Gamma_{\Delta}^{\mathrm{C,B}}$	$\widehat{\varphi}_{j\mathrm{C}}(\boldsymbol{x})$	P^{B}	Neumann boundary condition, Equation (3.28)
	$e_{i\mathrm{K}} \subset \Gamma_{\Delta}^{\mathrm{K,A}}$	$\widehat{\varphi}_{i\mathrm{K}}(\boldsymbol{x})$	P^{A}	Neumann boundary condition, Equation (3.28)
	$e_{j\mathrm{K}} \subset \Gamma_{\Delta}^{\mathrm{K,B}}$	$\widehat{\varphi}_{j\mathrm{K}}(\boldsymbol{x})$	P^{B}	Neumann boundary condition, Equation (3.28)

high-order accurate scheme is then derived based on the unconstrained and constrained polynomial reconstructions computed for the subproblems and, for those associated with the boundary edges, the ROD method is applied to properly fulfil the boundary conditions prescribed on the curved physical boundaries.

3.5.1 Finite volume method

The generic discretization of subproblems P^{S} with the finite volume method consists in applying the divergence theorem to the integral form of Equations (3.10) and (3.14) over each cell c_i in associated computational subdomain Ω_{Δ}^{S}, given as

$$\int_{\partial c_i} \left(\boldsymbol{u}^{S}(\boldsymbol{x})\phi^{S}(\boldsymbol{x}) - \kappa^{S}(\boldsymbol{x})\nabla\phi^{S}(\boldsymbol{x}) \right) \cdot \boldsymbol{s}_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_{c_i} f^{S}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{3.29}$$

where ∂c_i stands for the cell boundary and $\boldsymbol{s}_i(\boldsymbol{x})$ is the associated outward unit normal vector.

An R-points quadrature rule with weights ζ_r, $r = 1,\ldots,R$, is used to approximate the integrals on the straight edges and an Q-points quadrature rule with weights ξ_q, $q = 1,\ldots,Q$, is used to approximate the integrals in the polygonal cells. Gaussian quadrature rules are considered with weights that provide the approximate mean-values of the functions to evaluate. Given that, Equation (3.29) is rewritten in the discrete form in terms of numerical integrals, given as

$$\sum_{j\in\mathcal{N}_i} |e_{ij}| \left[\sum_{r=1}^{R} \zeta_r \left(C_{ij,r} + D_{ij,r} \right) \right] - f_i|c_i| = \mathcal{O}\left(h_i^{\alpha}\right), \tag{3.30}$$

where $h_i = |c_i|^{1/2}$ and α is the convergence order of the quadrature rule. Moreover, f_i stands for the

approximate cell mean-value of source term function $f^S(\boldsymbol{x})$ over cell c_i, given as

$$f_i = \sum_{r=1}^{Q} \xi_q f^S\left(\boldsymbol{q}_{i,q}\right), \tag{3.31}$$

and $C_{ij,r}$ and $D_{ij,r}$ stand for the physical convective and conductive heat fluxes, respectively, at the quadrature points of edge e_{ij}, given as

$$C_{ij,r} = \left(\boldsymbol{u}^S\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij}\right) \phi^S\left(\boldsymbol{q}_{ij,r}\right), \tag{3.32}$$

$$D_{ij,r} = -\kappa^S\left(\boldsymbol{q}_{ij,r}\right) \nabla \phi^S\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij}. \tag{3.33}$$

Notice that, for inner edge e_{ij} (the same as e_{ji}), the conservation of the convective and conductive heat fluxes is intrinsically preserved within the finite volume method, that is, $C_{ij,r} = -C_{ji,r}$ and $D_{ij,r} = -D_{ji,r}$, since the associated quadrature points coincide and the associated unit normal vectors are symmetric, that is, $\boldsymbol{q}_{ij,r} = \boldsymbol{q}_{ji,r}$ and $\boldsymbol{s}_{ij} = -\boldsymbol{s}_{ji}$.

Remark 14 To approximate the integrals in the polygonal cells, only Gaussian quadrature rules in triangular elements are necessary, even for meshes consisting of general polygonal cells. In the case of a cell with more than three sides, the polygon is split into triangles sharing the cell centroid as a common vertex, and the quadrature rules in triangular elements are applied in each triangular section. Then, the approximate cell mean-value corresponds to the summation of the mean-values in all sections weighted with the respective areas [75]. In that way, the approximate cell mean-values of the associated source term functions in general polygonal cells are determined without the need for specific quadrature rules for general polygons.

3.5.2 Very high-order accurate scheme

After the generic finite volume discretization of the subproblems, the very high-order accurate scheme consists in providing numerical approximations to physical convective and conductive heat fluxes $C_{ij,r}$ and $D_{ij,r}$, respectively. The numerical convective and conductive heat fluxes are denoted as $\mathcal{C}_{ij,r}$ and $\mathcal{D}_{ij,r}$, respectively, and are determined based on the computed unconstrained and constrained polynomial reconstructions for the associated subproblem (see Table 3.2).

3.5.2.1 Inner edges

For each inner edge e_{ij} in computational subdomain Ω_{Δ}^{S}, numerical convective and conductive heat fluxes $\mathcal{C}_{ij,r}$ and $\mathcal{D}_{ij,r}$, respectively, at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1, \ldots, R$, are given as

$$\mathcal{C}_{ij,r} = \left[\boldsymbol{u}^S\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij}\right]^{+} \widehat{\varphi}_i\left(\boldsymbol{q}_{ij,r}\right) + \left[\boldsymbol{u}^S\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij}\right]^{-} \widehat{\varphi}_j\left(\boldsymbol{q}_{ij,r}\right), \tag{3.34}$$

$$\mathcal{D}_{ij,r} = -\kappa^S\left(\boldsymbol{q}_{ij,r}\right)\nabla\widetilde{\varphi}_{ij}\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}, \tag{3.35}$$

where $[a]^+ = \max(0,a)$ and $[a]^- = \min(0,a)$ for any scalar $a \in \mathbb{R}$. Only polynomial reconstructions computed for the subproblem associated with the inner edge are used to determine the numerical approximations of the physical heat fluxes. For the numerical convective heat fluxes, the constrained polynomial reconstructions associated with the adjacent cells are used while for the numerical conductive heat fluxes the unconstrained polynomial reconstructions associated with the inner edges are used.

3.5.2.2 Boundary edges

For each boundary edge e_{iF} on computational boundary subset $\Gamma_\Delta^{F,S}$, numerical convective and conductive heat fluxes $\mathcal{C}_{iF,r}$ and $\mathcal{D}_{iF,r}$, respectively, at quadrature points $\boldsymbol{q}_{iF,r}$, $r = 1,\ldots,R$, are given as

$$\mathcal{C}_{iF,r} = \left[\boldsymbol{u}^S\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}\right]^+\widehat{\varphi}_i\left(\boldsymbol{q}_{iF,r}\right) + \left[\boldsymbol{u}^S\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}\right]^-\widehat{\varphi}_{iF}\left(\boldsymbol{q}_{iF,r}\right), \tag{3.36}$$

$$\mathcal{D}_{iF,r} = -\kappa^S\left(\boldsymbol{q}_{iF,r}\right)\nabla\widehat{\varphi}_{iF}\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}. \tag{3.37}$$

As previously, only polynomial reconstructions computed for the subproblem associated with the boundary edge are used to determined the numerical approximations of the physical heat fluxes. For the numerical convective heat fluxes, the constrained polynomial reconstructions associated with the adjacent cells are used while for the numerical conductive heat fluxes the constrained polynomial reconstructions associated with the boundary edges are used.

Notice that, for counterpart boundary edges e_{iF} and e_{jF} on computational boundary subsets $\Gamma_\Delta^{F,\mathrm{A}}$ and $\Gamma_\Delta^{F,\mathrm{B}}$, $F \in \{\mathrm{C},\mathrm{K}\}$, respectively, Equations (3.36) and (3.37) provide numerical convective and conductive heat fluxes that are not necessarily conservative, that is, $\mathcal{C}_{iF,r} + \mathcal{C}_{jF,r} \neq 0$ and $\mathcal{D}_{iF,r} + \mathcal{D}_{jF,r} \neq 0$ in the general case. Indeed, due to the geometrical mismatch between the computational and physical boundaries, nothing imposes that the numerical heat fluxes are conservative on the boundary edges. Moreover, although the counterpart boundary edges coincide geometrically, the associated numerical heat fluxes concern different subproblems with solutions that do not correspond (that is, no condition is imposed) on the computational boundaries. On the contrary, the conservation of the conductive heat fluxes is fulfilled on the physical boundaries following the ROD method to provide the appropriate linear constraints to compute the associated constrained polynomial reconstructions.

Notice that, the prescribed boundary conditions are taken into account to compute the constrained polynomial reconstructions associated to the boundary edges and, therefore, no explicit reference appears in the discretization scheme, which handles only two situations, inner or boundary edges.

Remark 15 A particular attention to the velocity, thermal conductivity, and source term is required in the case of non-convex curved physical boundaries. In Equations (3.36) and (3.37), the numerical convective and conductive heat fluxes are determined evaluating functions $\boldsymbol{u}^S(\boldsymbol{x})$ and $\kappa^S(\boldsymbol{x})$, respectively, at the quadrature points of the boundary edges in the associated computational physical boundary. In the case of concave curved physical boundaries, the associated computational physical boundary is composed of boundary edges necessarily located outside the physical domain and, therefore, functions $\boldsymbol{u}^S(\boldsymbol{x})$ and $\kappa^S(\boldsymbol{x})$ do not have a physical meaning at the edge quadrature points. Similarly, in Equation (3.31), the approximate cell mean-values of the source term functions are determined evaluating functions $\boldsymbol{f}^S(\boldsymbol{x})$ at the quadrature points of the cells in the associated computational subdomain. In the case of concave curved physical boundaries, the cells adjacent to the boundary edges cover the geometrical mismatch between the physical and the computational domains. Therefore, some of their quadrature points are eventually located outside the physical domain (with low-order accurate quadrature rules this situation does not necessarily happen), where functions $f^S(\boldsymbol{x})$ do not have a physical meaning. To overcome this issue, representative extensions of functions $\boldsymbol{u}^S(\boldsymbol{x})$, $\kappa^S(\boldsymbol{x})$, and $f^S(\boldsymbol{x})$ are necessary at the quadrature points of the boundary edges and cells located outside the physical domain. In the case of given functions, the provided expression is used to extrapolate the values at the quadrature points and, even not having a physical meaning, the technique preserves the accuracy and the convergence order of the proposed method. In the case of functions that are variables of the problem (for instance, the velocity) approximate cell mean-values are used to compute a local reconstruction and to extrapolate the values at the quadrature points.

Remark 16 Consider counterpart boundary edges e_{iF} and e_{jF} on computational boundary subsets $\Gamma_\Delta^{F,A}$ and $\Gamma_\Delta^{F,B}$, $F \in \{C,K\}$, respectively. The prescribed continuity and imperfect interface conditions impose the conservation of the conductive heat flux on the interface and the convective heat flux through the interface is null. Therefore, the numerical conductive heat fluxes at the quadrature points of one boundary edge are determined and the same values are assigned to those associated with the counterpart boundary edge, that is, $\mathcal{D}_{iF,r} = -\mathcal{D}_{jF,r} = -\kappa^A(\boldsymbol{q}_{iF,r})\nabla\widehat{\varphi}_{iF}(\boldsymbol{q}_{iF,r})\cdot\boldsymbol{s}_{iF}$. Moreover, since there is no convective heat flux through the interface, numerical convective heat fluxes at the quadrature points of both boundary edges are assigned to null, that is, $\mathcal{C}_{iF,r} = \mathcal{C}_{jF,r} = 0$. Notice that this discretization scheme does not require polynomial reconstruction $\widehat{\varphi}_{jF}(\boldsymbol{x})$ computed for subproblem P^B where the Neumann boundary conditions (3.16) and (3.17) are prescribed to conserve the conductive heat flux. Prescribing symmetric numerical conductive heat fluxes, that is, $\mathcal{D}_{jF,r} = -\mathcal{D}_{iF,r}$, seems to intrinsically satisfy the conservation condition, at least, on the computational boundaries. However, for curved physical domains, physical boundary subsets $\Gamma^{F,S}$ and associated computational boundary subsets $\Gamma_\Delta^{F,S}$ do not fully overlap, and the geometrical mismatch implies a deterioration of the convergence order of the method with this discretization scheme. Similarly, normal velocity $\boldsymbol{u}^S(\boldsymbol{x})\cdot\boldsymbol{n}^S(\boldsymbol{x})$ vanishes on the curved physical boundaries, but the numerical convective heat fluxes are not generally null on the associated computational boundaries. Hence, this discretization scheme only provides the optimal convergence order of the method in the case of polygonal domains.

3.5.3 Thermal coupling

Boundary condition functions $g^{\mathrm{C,A}}(\boldsymbol{x})$ and $g^{\mathrm{C,B}}(\boldsymbol{x})$ assigned for the thermal coupling are given in terms of temperature functions $\phi^{\mathrm{A}}(\boldsymbol{x})$ and $\phi^{\mathrm{B}}(\boldsymbol{x})$. Therefore, associated point-values at the collocation points need to be determined as approximations of associated boundary condition functions, and the method consists in using the same polynomial reconstructions used to determine the numerical approximations of the physical heat fluxes.

3.5.3.1 Boundary edges on continuity computational boundary subset

Consider counterpart boundary edges $e_{i\mathrm{C}}$ and $e_{j\mathrm{C}}$ on computational boundary subsets $\Gamma_{\Delta}^{\mathrm{C,A}}$ and $\Gamma_{\Delta}^{\mathrm{C,B}}$, respectively. Therefore, cell c_i belongs to computational subdomain $\Omega_{\Delta}^{\mathrm{A}}$ and cell c_j belongs to computational subdomain $\Omega_{\Delta}^{\mathrm{B}}$.

From the Dirichlet-Neumann method, the thermal coupling between subproblems to recover the continuity interface conditions consists in assigning functions $g^{\mathrm{C,A}}(\boldsymbol{x}) := \phi^{\mathrm{B}}(\boldsymbol{x})$ and $g^{\mathrm{C,B}}(\boldsymbol{x}) := -\kappa^{\mathrm{A}}(\boldsymbol{x})\nabla\phi^{\mathrm{A}}(\boldsymbol{x})\cdot\boldsymbol{n}^{\mathrm{B}}(\boldsymbol{x})$. Therefore, point-values $g_{i\mathrm{C}}$ and $g_{j\mathrm{C}}$ are determined with the evaluation of polynomial reconstructions $\widehat{\varphi}_j(\boldsymbol{x})$ and $\widehat{\varphi}_{iC}(\boldsymbol{x})$ at collocation points $\boldsymbol{p}_{i\mathrm{C}} = \boldsymbol{p}_{j\mathrm{C}}$ with associated unit normal vectors $\boldsymbol{v}_{i\mathrm{C}} = -\boldsymbol{v}_{j\mathrm{C}}$ belonging to the associated physical boundary subsets, given as

$$g_{i\mathrm{C}} = \widehat{\varphi}_j\left(\boldsymbol{p}_{i\mathrm{C}}\right), \tag{3.38}$$

$$g_{j\mathrm{C}} = -\kappa^{\mathrm{A}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\nabla\widehat{\varphi}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\cdot\boldsymbol{v}_{j\mathrm{C}}. \tag{3.39}$$

Polynomial reconstruction $\widehat{\varphi}_{i\mathrm{C}}(\boldsymbol{x})$ for subproblem P^{A} is constrained with boundary variable $g_{i\mathrm{C}}$ determined with Equation (3.38) from constrained polynomial reconstruction $\widehat{\varphi}_j(\boldsymbol{x})$ for subproblem P^{B}. In turn, polynomial reconstruction $\widehat{\varphi}_{j\mathrm{C}}(\boldsymbol{x})$ for subproblem P^{B} is constrained with boundary variable $g_{j\mathrm{C}}$ determined with Equation (3.39) from constrained polynomial reconstruction $\widehat{\varphi}_{i\mathrm{C}}(\boldsymbol{x})$ for subproblem P^{A}. Therefore, the data of subproblem P^{B} is used to approximate the prescribed Dirichlet boundary condition function in subproblem P^{A}. In contrast, the data of the latter is used to approximate the prescribed Neumann boundary condition function in subproblem P^{B}.

Remark 17 Alternatively, boundary variable g_{jC} can be determined using the constrained polynomial reconstruction for the cell mean-value conservation in subproblem P^{A}, given as $g_{jC} = -\kappa^{A}\left(\boldsymbol{p}_{jC}\right)\nabla\widehat{\varphi}_i\left(\boldsymbol{p}_{jC}\right)\cdot\boldsymbol{v}_{jC}$. In numerical experiments, although constrained polynomial reconstructions for the cell mean-value conservation are appropriate to determine an approximation of Dirichlet boundary condition functions, an accuracy deterioration is obtained in the case of Neumann boundary condition functions. Therefore, the optimal convergence order of the method is not preserved.

3.5.3.2 Boundary edges on Kapitza computational boundary subset

Consider counterpart boundary edges $e_{i\mathrm{K}}$ and $e_{j\mathrm{K}}$ on computational boundary subsets $\Gamma_{\Delta}^{\mathrm{K,A}}$ and $\Gamma_{\Delta}^{\mathrm{K,B}}$, respectively. Therefore, cell c_i belongs to computational subdomain $\Omega_{\Delta}^{\mathrm{A}}$ and cell c_j belongs to computational subdomain $\Omega_{\Delta}^{\mathrm{B}}$.

From the Neumann-Neumann method, the thermal coupling between subproblems to recover the imperfect interface conditions consists in assigning functions $g^{\mathrm{K,A}}(\boldsymbol{x}) := h^{\mathrm{K}}(\boldsymbol{x})\big(\phi^{\mathrm{A}}(\boldsymbol{x}) - \phi^{\mathrm{B}}(\boldsymbol{x})\big)$ and $g^{\mathrm{K,B}}(\boldsymbol{x}) := h^{\mathrm{K}}(\boldsymbol{x})\big(\phi^{\mathrm{B}}(\boldsymbol{x}) - \phi^{\mathrm{A}}(\boldsymbol{x})\big)$. Therefore, boundary variables $g_{i\mathrm{K}}$ and $g_{j\mathrm{K}}$ are determined with the evaluation of the polynomial reconstructions $\widehat{\varphi}_i(\boldsymbol{x})$ and $\widehat{\varphi}_j(\boldsymbol{x})$ at collocation points $\boldsymbol{p}_{i\mathrm{K}} = \boldsymbol{p}_{j\mathrm{K}}$ belonging to the associated physical boundary subsets, given as

$$g_{i\mathrm{K}} = h^{\mathrm{K}}(\boldsymbol{p}_{i\mathrm{K}})\big(\widehat{\varphi}_i(\boldsymbol{p}_{i\mathrm{K}}) - \widehat{\varphi}_j(\boldsymbol{p}_{i\mathrm{K}})\big), \tag{3.40}$$

$$g_{j\mathrm{K}} = h^{\mathrm{K}}\Big(\boldsymbol{p}_{j\mathrm{K}}\Big)\Big(\widehat{\varphi}_j\Big(\boldsymbol{p}_{j\mathrm{K}}\Big) - \widehat{\varphi}_i\Big(\boldsymbol{p}_{j\mathrm{K}}\Big)\Big), \tag{3.41}$$

where the symmetry $g_{j\mathrm{K}} = -g_{i\mathrm{K}}$ is noticed.

Polynomial reconstructions $\widehat{\varphi}_{i\mathrm{K}}(\boldsymbol{x})$ and $\widehat{\varphi}_{j\mathrm{K}}(\boldsymbol{x})$ for subproblems P^{A} and P^{B}, respectively, are constrained with boundary variables $g_{i\mathrm{K}}$ and $g_{j\mathrm{K}}$, respectively, determined with Equations (3.40) and (3.41) from constrained polynomial reconstructions $\widehat{\varphi}_i(\boldsymbol{x})$ and $\widehat{\varphi}_j(\boldsymbol{x})$. Therefore, the data from subproblems P^{A} and P^{B} is used to approximate the prescribed Neumann boundary condition functions in both subproblems.

3.5.4 Residual operators

The cell mean-values of temperature functions $\phi^S(\boldsymbol{x})$ are gathered in vectors $\boldsymbol{\Phi}^S \in \mathbb{R}^{n^S}$ and the point-values of boundary condition functions $g^{F,S}(\boldsymbol{x})$ are gathered in vectors $\boldsymbol{\Psi}^{F,S} \in \mathbb{R}^{n^{F,S}}$, that is, $\boldsymbol{\Phi}^S = [\phi_i]_{i\in\mathcal{I}^S}$ and $\boldsymbol{\Psi}^{F,S} = [g_{iF}]_{i\in\mathcal{I}^{F,S}}$. The equations derived from proposed method, that is, the discretization of the subproblems and the numerical treatment of the thermal coupling, are rewritten in terms of affine residual operators, as follows:

- For each cell c_i in computational subdomain Ω_{Δ}^S, Equation (3.30) derived from the discretization of the subproblems is rewritten with the numerical heat fluxes as $\mathcal{R}_i = 0$ with residual given as

$$\mathcal{R}_i = \sum_{j\in\mathcal{N}_i} |e_{ij}| \left[\sum_{r=1}^{R} \zeta_r \big(\mathcal{C}_{ij,r} + \mathcal{D}_{ij,r}\big) \right] - f_i|c_i|, \tag{3.42}$$

  where the cell mean-values in vector $\boldsymbol{\Phi}^S$ and the point-values in vectors $\boldsymbol{\Psi}^{F,S}$ are necessary to compute the associated polynomial reconstructions. The residuals defined in Equation (3.42) for cells c_i in computational subdomain Ω_{Δ}^S with only inner faces are gathered in the vector form with affine residual operator $\mathcal{R}^S\big(\boldsymbol{\Phi}^S\big)$ from $\mathbb{R}^{n^S}$ to $\mathbb{R}^{n^S - n^{\mathrm{D},S} - n^{\mathrm{C},S} - n^{\mathrm{K},S}}$. The residuals defined in

Equation (3.42) for boundary cells c_i in computational subdomain Ω_{Δ}^{S} with a boundary edge e_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$ are gathered in the vector form with affine residual operator $\mathcal{R}^{F,S}\left(\boldsymbol{\Phi}^{S}, \boldsymbol{\Psi}^{F,S}\right)$ from $\mathbb{R}^{n^{S}+n^{F,S}}$ to $\mathbb{R}^{n^{F,S}}$.

- For each boundary edge $e_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,A}}$ and counterpart boundary edge $e_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$, Equation (3.38) and (3.39) derived from the numerical treatment of the thermal coupling are rewritten as $\mathcal{R}_{i\mathrm{G}} = 0$ and $\mathcal{G}_{j\mathrm{C}} = 0$, respectively, with residuals given as

$$\mathcal{G}_{i\mathrm{C}} = \widehat{\varphi}_{j}\left(\boldsymbol{p}_{i\mathrm{C}}\right) - g_{i\mathrm{C}}, \tag{3.43}$$

$$\mathcal{G}_{j\mathrm{C}} = -\kappa^{\mathrm{A}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\nabla\widehat{\varphi}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\cdot\boldsymbol{\nu}_{j\mathrm{C}} - g_{j\mathrm{C}}, \tag{3.44}$$

where the cell mean-values in vectors $\boldsymbol{\Phi}^{\mathrm{B}}$ are necessary to compute the associated polynomial reconstructions in the former case while the cell mean-values in vectors $\boldsymbol{\Phi}^{\mathrm{A}}$ and the point-values in vector $\boldsymbol{\Psi}^{\mathrm{C,A}}$ are necessary to compute the associated polynomial reconstructions in the latter case. The residuals defined in Equation (3.43) for boundary edge $e_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,A}}$ are gathered in the vector form with affine residual operator $\mathcal{G}^{\mathrm{C,A}}\left(\boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)$ from $\mathbb{R}^{n^{\mathrm{C,A}}+n^{\mathrm{B}}}$ to $\mathbb{R}^{n^{\mathrm{C,A}}}$. The residuals defined in Equation (3.44) for counterpart boundary edge $e_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$ are gathered in the vector form with affine residual operator $\mathcal{G}^{\mathrm{C,B}}\left(\boldsymbol{\Psi}^{\mathrm{C,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right)$ from $\mathbb{R}^{n^{\mathrm{C,B}}+n^{\mathrm{A}}+n^{\mathrm{C,A}}}$ to $\mathbb{R}^{n^{\mathrm{C,B}}}$.

- For each boundary edge $e_{i\mathrm{K}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{K,A}}$ and counterpart boundary edge $e_{j\mathrm{K}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{K,B}}$, Equations (3.40) and (3.41) derived from the numerical treatment of the thermal coupling is rewritten as $\mathcal{R}_{i\mathrm{G}} = 0$ and $\mathcal{G}_{j\mathrm{K}} = 0$, respectively, with residuals given as

$$\mathcal{G}_{i\mathrm{K}} = h^{\mathrm{K}}(\boldsymbol{p}_{i\mathrm{K}})\left(\widehat{\varphi}_{i}(\boldsymbol{p}_{i\mathrm{K}}) - \widehat{\varphi}_{j}(\boldsymbol{p}_{i\mathrm{K}})\right) - g_{i\mathrm{K}}, \tag{3.45}$$

$$\mathcal{G}_{j\mathrm{K}} = h^{\mathrm{K}}\left(\boldsymbol{p}_{j\mathrm{K}}\right)\left(\widehat{\varphi}_{j}\left(\boldsymbol{p}_{j\mathrm{K}}\right) - \widehat{\varphi}_{i}\left(\boldsymbol{p}_{j\mathrm{K}}\right)\right) - g_{j\mathrm{K}}, \tag{3.46}$$

where the cell mean-values in vectors $\boldsymbol{\Phi}^{\mathrm{A}}$ and $\boldsymbol{\Phi}^{\mathrm{B}}$ are necessary to compute the associated polynomial reconstructions in both cases. The residuals defined in Equation (3.45) for boundary edge $e_{i\mathrm{K}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{K,A}}$ are gathered in the vector form with affine residual operator $\mathcal{G}^{\mathrm{K,A}}\left(\boldsymbol{\Psi}^{\mathrm{K,A}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)$ from $\mathbb{R}^{n^{\mathrm{K,A}}+n^{\mathrm{A}}+n^{\mathrm{B}}}$ to $\mathbb{R}^{n^{\mathrm{K,A}}}$. The residuals defined in Equation (3.46) for counterpart boundary edge $e_{j\mathrm{K}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{K,B}}$ are gathered in the vector form with affine residual operator $\mathcal{G}^{\mathrm{K,B}}\left(\boldsymbol{\Psi}^{\mathrm{K,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)$ from $\mathbb{R}^{n^{\mathrm{K,B}}+n^{\mathrm{A}}+n^{\mathrm{B}}}$ to $\mathbb{R}^{n^{\mathrm{K,B}}}$.

3.5.5 Implicit formulation

The cell mean-values of temperature functions $\phi^{S}(\boldsymbol{x})$ in vectors $\boldsymbol{\Phi}^{S}$ are unknown and correspond to the internal variables for the proposed method while the point-values of boundary condition functions $g^{\mathrm{C},S}(\boldsymbol{x})$ and $g^{\mathrm{K},S}(\boldsymbol{x})$ in vectors $\boldsymbol{\Psi}^{\mathrm{C},S}$ and $\boldsymbol{\Psi}^{\mathrm{K},S}$, respectively, are also unknown but correspond to the boundary variables for the proposed method. Notice that boundary conditions functions $g^{\mathrm{D},S}(\boldsymbol{x})$ are given and, therefore, vectors $\boldsymbol{\Psi}^{\mathrm{D},S}$ gather, in fact, only known point-values. Therefore, the number of degrees of freedom to determine is $DOF = n^{\mathrm{A}} + n^{\mathrm{B}} + n^{\mathrm{C,A}} + n^{\mathrm{C,B}} + n^{\mathrm{K,A}} + n^{\mathrm{K,B}}$. The internal and boundary variables for the proposed method are gathered in global vector $\boldsymbol{\Phi} \in \mathbb{R}^{DOF}$, given as $\boldsymbol{\Phi} = \left[\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}, \boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Psi}^{\mathrm{C,B}}, \boldsymbol{\Psi}^{\mathrm{K,A}}, \boldsymbol{\Psi}^{\mathrm{K,B}}\right]^{\mathrm{T}}$.

The implicit system of linear equations for the solution of the partitioned problem is formed gathering affine residual operators (3.42) and (3.46) in vector form in global affine residual operator $\mathcal{H}(\boldsymbol{\Phi})$ from $\mathbb{R}^{DOF}$ to $\mathbb{R}^{DOF}$, given as

$$\mathcal{H}(\boldsymbol{\Phi}) = \begin{bmatrix} \mathcal{R}^{\mathrm{A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}\right) \\ \mathcal{R}^{\mathrm{D,A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{D,A}}\right) \\ \mathcal{R}^{\mathrm{C,A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right) \\ \mathcal{R}^{\mathrm{K,A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{K,A}}\right) \\ \mathcal{R}^{\mathrm{B}}\left(\boldsymbol{\Phi}^{\mathrm{B}}\right) \\ \mathcal{R}^{\mathrm{D,B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{D,B}}\right) \\ \mathcal{R}^{\mathrm{C,B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,B}}\right) \\ \mathcal{R}^{\mathrm{K,B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{K,B}}\right) \\ \mathcal{G}^{\mathrm{C,A}}\left(\boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) \\ \mathcal{G}^{\mathrm{C,B}}\left(\boldsymbol{\Psi}^{\mathrm{C,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right) \\ \mathcal{G}^{\mathrm{K,A}}\left(\boldsymbol{\Psi}^{\mathrm{K,A}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) \\ \mathcal{G}^{\mathrm{K,B}}\left(\boldsymbol{\Psi}^{\mathrm{K,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) \end{bmatrix}, \tag{3.47}$$

The solution of the implicit system of linear equations given as $\mathcal{H}(\boldsymbol{\Phi}) = \boldsymbol{0}$ provides vector $\boldsymbol{\Phi}^{*} \in \mathbb{R}^{DOF}$ that corresponds to the approximate solution of the conjugate heat transfer problem in terms of internal and boundary variables. In that case, the internal variables correspond to the approximate cell mean-values of temperature functions $\phi^{S}(\boldsymbol{x})$ in the cells of the mesh and the boundary variables correspond to the approximate point-values of boundary condition functions $g^{F,S}(\boldsymbol{x})$, $F \in \{\mathrm{C}, \mathrm{K}\}$, at the boundary collocation points. An iterative free-matrix algorithm, such as the GMRES method, supplemented with a preconditioning matrix, is used to solve the system of linear equations, as proposed in Clain et al., 2013 [66], and Bouralas et al., 2017 [63]. The free-matrix

approach avoids to explicitly assemble and store the matrix of the coefficients derived from the residual operators, which requires algorithms that are not elaborated in the present work.

Remark 18 An alternative solution method consists in an iterative algorithm that solves apart and successively subproblems P^S for the internal variables, such that the solution of systems of linear equations $\mathcal{R}^S\left(\boldsymbol{\Phi}^S, \boldsymbol{\Psi}^{C,S}, \boldsymbol{\Psi}^{K,S}\right) = \boldsymbol{0}$ with the residual given as

$$\mathcal{H}^S\left(\boldsymbol{\Phi}^S, \boldsymbol{\Psi}^{C,S}, \boldsymbol{\Psi}^{K,S}\right) = \begin{bmatrix} \mathcal{R}^S\left(\boldsymbol{\Phi}^S\right) \\ \mathcal{R}^{D,S}\left(\boldsymbol{\Phi}^S, \boldsymbol{\Psi}^{D,S}\right) \\ \mathcal{R}^{C,S}\left(\boldsymbol{\Phi}^S, \boldsymbol{\Psi}^{C,S}\right) \\ \mathcal{R}^{K,S}\left(\boldsymbol{\Phi}^S, \boldsymbol{\Psi}^{K,S}\right) \end{bmatrix} \tag{3.48}$$

provides a sequence of vectors $\boldsymbol{\Phi}^{S,*} \in \mathbb{R}^{n^S}$ that converge to the solution of the conjugate heat transfer problem. This also requires that, between two consecutive iterations, boundary variables in vectors $\boldsymbol{\Phi}^{C,S,*} \in \mathbb{R}^{n^{C,S}}$ and $\boldsymbol{\Phi}^{K,S,*} \in \mathbb{R}^{n^{C,S}}$ are determined from the previous intermediate internal variables in vectors $\boldsymbol{\Phi}^{S,*}$ using the associated residual operators (no system of linear equations is solved to determine the boundary variables from given internal variables). This method often converges slower than the direct solution method elaborated previously, where the speed of convergence is closely related with the quality of the initial guess solution (and, for that reason, initial coarse solvers are usually applied) and the strength of the thermal interaction between subproblems.

3.5.6 Static condensation

Since the numerical treatment of the thermal coupling with Equations (3.38) and (3.41) is independent for each boundary variable of the same subproblem, explicit expressions for vectors $\boldsymbol{\Psi}^{F,S}$ as functions of only vectors $\boldsymbol{\Phi}^S$ are possible to derive, such that

$$\mathcal{G}^{\mathrm{C,A}}\left(\boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) = \boldsymbol{0}, \tag{3.49}$$

$$\mathcal{G}^{\mathrm{C,B}}\left(\boldsymbol{\Psi}^{\mathrm{C,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right) = \boldsymbol{0}, \tag{3.50}$$

$$\mathcal{G}^{\mathrm{K,A}}\left(\boldsymbol{\Psi}^{\mathrm{K,A}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) = \boldsymbol{0}, \tag{3.51}$$

$$\mathcal{G}^{\mathrm{K,B}}\left(\boldsymbol{\Psi}^{\mathrm{K,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) = \boldsymbol{0}. \tag{3.52}$$

In that way, there are affine functions $\mathcal{G}'^{,F,S}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)$ from $\mathbb{R}^{n^{\mathrm{A}}+n^{\mathrm{B}}}$ to $\mathbb{R}^{n^{F,S}}$ that provide vectors $\boldsymbol{\Psi}^{F,S}$ explicitly, given as

$$\boldsymbol{\Psi}^{\mathrm{C,A}} = \mathcal{G}'^{,\mathrm{C,A}}\left(\boldsymbol{\Phi}^{\mathrm{B}}\right), \tag{3.53}$$

$$\boldsymbol{\Psi}^{\mathrm{C,B}} = \mathcal{G}'^{,\mathrm{C,B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right), \tag{3.54}$$

$$\boldsymbol{\Psi}^{\mathrm{K,A}} = \mathcal{G}'^{,\mathrm{K,A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right), \tag{3.55}$$

$$\boldsymbol{\Psi}^{\mathrm{K,B}} = \mathcal{G}'^{,\mathrm{K,B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right), \tag{3.56}$$

with the particular case of operator $\mathcal{G}'^{,\mathrm{C,A}}\left(\boldsymbol{\Phi}^{\mathrm{B}}\right)$ from $\mathbb{R}^{n^{\mathrm{B}}}$ to $\mathbb{R}^{n^{\mathrm{C,A}}}$, which does not depend on vector $\boldsymbol{\Phi}^{\mathrm{A}}$ by construction.

The global residual operator (3.47) is statically condensed and given in terms of only vectors $\boldsymbol{\Phi}^{S}$ eliminating the rows associated with vectors $\boldsymbol{\Psi}^{F,S}$. Therefore, consider that the internal variables for both subproblems are gathered in vector $\boldsymbol{\Phi}' \in \mathbb{R}^{n^{\mathrm{A}}+n^{\mathrm{B}}}$, given as $\boldsymbol{\Phi}' = \left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)$, whereas no boundary variables are considered. Therefore, the number of degrees of freedom to determine is $DOF = n^{\mathrm{A}} + n^{\mathrm{B}} + n^{\mathrm{C,A}}$.

The condensed system of linear equations for the solution of the partitioned problem is formed rewriting affine residual operators $\mathcal{H}^{S}\left(\boldsymbol{\Phi}^{S}, \boldsymbol{\Psi}^{\mathrm{C},S}, \boldsymbol{\Psi}^{\mathrm{K},S}\right)$ as affine residual operators $\mathcal{H}'^{,S}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)$ from $\mathbb{R}^{n^{\mathrm{A}}+n^{\mathrm{B}}}$ to $\mathbb{R}^{n^{S}}$, given as

$$\mathcal{H}'^{,\mathrm{A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) = \mathcal{H}^{\mathrm{A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \mathcal{G}'^{,\mathrm{C,A}}\left(\boldsymbol{\Phi}^{\mathrm{B}}\right), \mathcal{G}'^{,\mathrm{K,A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)\right), \tag{3.57}$$

$$\mathcal{H}'^{,\mathrm{B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) = \mathcal{H}^{\mathrm{B}}\left(\boldsymbol{\Phi}^{\mathrm{B}}, \mathcal{G}'^{,\mathrm{C,B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right), \mathcal{G}'^{,\mathrm{K,B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)\right), \tag{3.58}$$

and, therefore, the statically condensed global affine residual operator $\mathcal{H}'(\boldsymbol{\Phi}')$ from $\mathbb{R}^{DOF}$ to $\mathbb{R}^{DOF}$ is given as

$$\mathcal{H}'(\boldsymbol{\Phi}') = \begin{bmatrix} \mathcal{H}'^{,\mathrm{A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) \\ \mathcal{H}'^{,\mathrm{B}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) \end{bmatrix}. \tag{3.59}$$

Given that, the solution of the statically condensed system of linear equations given as $\mathcal{H}'(\boldsymbol{\Phi}') = \boldsymbol{0}$ provides vector $\boldsymbol{\Phi}'^{,*} \in \mathbb{R}^{DOF}$ that corresponds to the approximate solution of the conjugate heat transfer problem in terms of internal variables. The static condensation method allows reducing the size of the system of linear equations to $n^{\mathrm{A}} + n^{\mathrm{B}}$. Contrarily to other static condensation methods, no additional problem is previously solved in terms of the boundary variables to provide the boundary data for the subproblems.

3.6 Numerical benchmark

The verification of the proposed method is based on the assessment of its numerical performance in terms of accuracy, convergence order, stability, and robustness to solve conjugate heat transfer

problems in arbitrary two-dimensional curved domains. For that purpose, specific analytic solutions provided with temperature functions $\phi^S(\boldsymbol{x})$ are obtained in associated physical subdomains Ω^S such that the prescribed continuity and imperfect interface conditions are fulfilled on interface subsets Γ^{C} and Γ^{K}, respectively. Moreover, functions $g^{\mathrm{D},S}(\boldsymbol{x})$ are obtained from the analytic solutions to prescribe the Dirichlet boundary conditions on physical boundary subsets $\Gamma^{\mathrm{D},S}$. Following the method of manufactured solutions [76–79], source term functions $f^S(\boldsymbol{x})$ are determined to satisfy Equation (3.1) in associated physical subdomains Ω^S given the analytic solution, the velocity vector functions $\boldsymbol{u}^S(\boldsymbol{x})$, and the thermal conductivity functions $\kappa^S(\boldsymbol{x})$. Notice that only smooth solutions and source terms of the steady-state three-dimensional conjugate heat transfer problem are considered in the present work. The method of verification consists in running simulations with successive finer polygonal meshes generated for the curved domains. The approximate solutions computed from the proposed method are compared with the analytic solutions and, therefore, the accuracy and convergence orders under mesh refinement are determined, as follows.

The system of linear equations is solved in the condensed form with the GMRES method where the number of degrees of freedom is $DOF = n^{\mathrm{A}} + n^{\mathrm{B}}$ corresponding to the approximate cell mean-values of the temperature functions. For each cell c_i in computational subdomain Ω^S_Δ, the error of approximate cell mean-value ϕ_i^* is determined having the exact cell mean-value given as

$$\overline{\phi}_i = \frac{1}{|c_i|}\int_{c_i} \phi^S(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{3.60}$$

Then, the relative errors in the L^1- and L^∞-norm, denoted as E_1 and E_∞, respectively, are determined as

$$E_1 = \frac{\sum_{i=1}^{DOF} \left|\phi_i^* - \overline{\phi}_i\right| |c_i|}{\sum_{i=1}^{DOF} |c_i|}, \tag{3.61}$$

$$E_\infty = \max_{i=1}^{DOF} \left|\phi_i^* - \overline{\phi}_i\right|. \tag{3.62}$$

Consider two meshes for the same domain with different characteristic sizes, a number of degrees of freedom of DOF_1 and DOF_2, respectively, associated relative errors in the L^1-norm of $E_{1,1}$ and $E_{1,2}$, respectively, and associated relative errors in the L^∞-norm of $E_{\infty,1}$ and $E_{\infty,2}$, respectively. Then, the convergence orders for the relative errors in the L^1- and L^∞-norms, denoted as O_1 and O_∞, respectively, are given as

$$O_1 = 2\left|\frac{\ln\left(E_{1,1}/E_{1,2}\right)}{\ln(DOF_1/DOF_2)}\right|, \tag{3.63}$$

$$O_{\infty} = 2\left|\frac{\ln\left(E_{\infty,1}/E_{\infty,2}\right)}{\ln(DOF_1/DOF_2)}\right|. \tag{3.64}$$

Remark 19 Contrarily to the common practice in elliptic problems, the relative errors are not measured in the typical L^2-norm but rather in the L^1- and L^∞-norms. Firstly, the measured relative errors in the L^k-norm, with $1 < k < \infty$, are within lower and upper limits corresponding to the relative errors measured in the L^1- and L^∞-norms, respectively. On the other side, the combination of these two norms provides a better perception of the obtained relative error distribution in the computational domain. Indeed, the presence of localized suspicious errors cannot be concluded from the L^1-norm, whereas the evenness of the relative error distribution cannot be deduced from the L^∞-norm. On the other hand, having both norms, large ratios between the L^1- and L^∞-norms indicate that a small number of cells have considerably larger errors than most of the remaining. For these reasons, the combined use of the L^1- and L^∞-norms is preferred to provide a better numerical assessment of the proposed method than the typical L^2-norm.

Variations of the proposed method, introducing some conventional and naive techniques, are considered to illustrate the full benefit of the undertaken development to solve conjugate heat transfer problems in curved domains. The only differences are as follows:

- Method 1: the method for polygonal domains in Remark 13 is used to determine the collocation points and the unit vectors in the linear constraints of the minimization procedure for the constrained polynomial reconstructions associated only with the boundary edges on computational boundary subsets $\Gamma_{\Delta}^{F,S}$, $F \in \{\mathrm{C},\mathrm{K}\}$, discretizing the interface.
- Method 2: the discretization scheme for polygonal domains in Remark 16 is used to determine the numerical convective and conductive heat fluxes associated only with the boundary edges on computational boundary subsets $\Gamma_{\Delta}^{F,S}$, $F \in \{\mathrm{C},\mathrm{K}\}$, discretizing the interface.

The proposed method is referred to as Method 3, for which the preliminary tests showed that accuracy and convergence orders improve with slight modifications. Therefore, in addition to the previous methods, the following modification to Method 3 is also considered:

- Method 4: degree $d+1$ is considered for the polynomial reconstructions associated only with the boundary edges on computational boundary subsets $\Gamma_{\Delta}^{\mathrm{C,A}}$, $\Gamma_{\Delta}^{\mathrm{K,A}}$, and $\Gamma_{\Delta}^{\mathrm{K,B}}$, discretizing the interface (with a Neumann boundary condition prescribed).
- Method 5: degree $d+1$ is considered for the polynomial reconstructions associated only with boundary edges on computational boundary subsets $\Gamma_{\Delta}^{F,S}$, $F \in \{\mathrm{C},\mathrm{K}\}$, discretizing the interface (with either a Dirichlet or a Neumann boundary conditions prescribed).

In Methods 4 and 5, the accuracy is enhanced at the cost of a small efficiency penalty due to the additional coefficients for the polynomial reconstructions of degree $d+1$ instead of d. In practice, this

penalty is small due to the relative number of polynomial reconstructions associated to the boundary edges, which are less than 5% of the total number of polynomial reconstructions computed in the test cases presented in this benchmark study.

In the case of the continuity interface conditions, the partitioning method prescribes either a Dirichlet boundary condition in one subproblem and a Neumann boundary condition in the other, resulting in two possible partitioning configurations (see Remark 12). These two partitioning configurations are referred to as configurations DN and ND and are defined as follows:

- Configuration DN: a Neumann boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,A}}$ in subproblem P^{A} and a Dirichlet boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,B}}$ in subproblem P^{B}.

- Configuration ND: a Dirichlet boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,A}}$ in subproblem P^{A} and a Neumann boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,B}}$ in subproblem P^{B}.

Specific test cases are addressed with both partitioning configurations, and the obtained approximate solutions are compared in terms of accuracy and convergence orders, establishing their behaviour according to the physical properties of the problem.

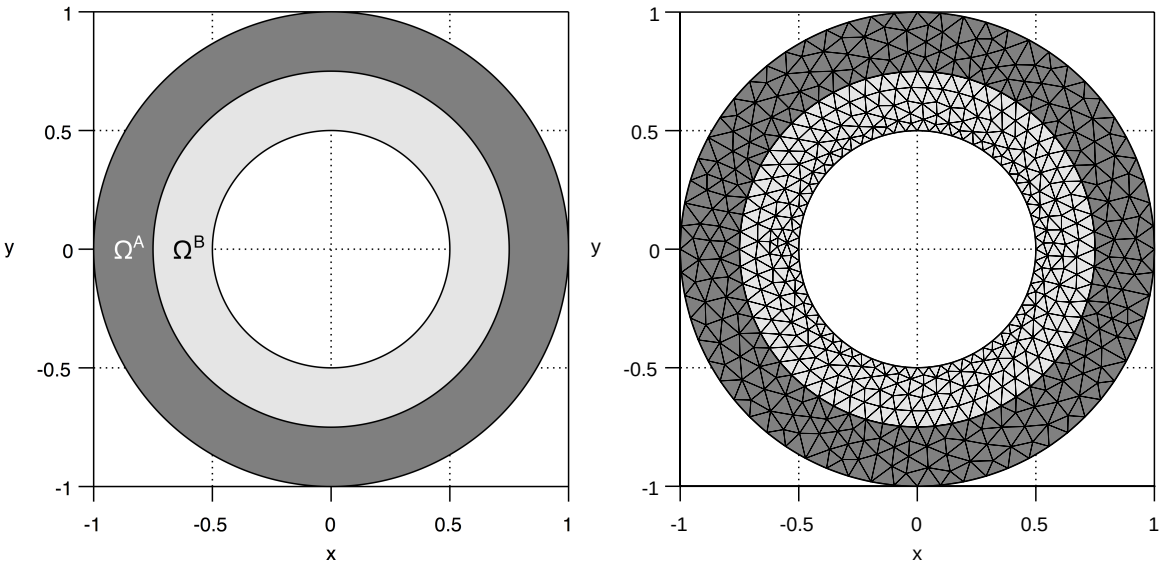
The weighting function to perform the minimization procedure in the polynomial reconstruction method is considered with parameters $\delta = 2$ and $\sigma = 5h$, where h is the characteristic size of the reference mesh element (face or cell). The optimal parameters are chosen to provide stability and robustness to the least-squares method.

3.6.1 Circular interface with the continuity interface conditions test case

An annular domain Ω is considered consisting of an external and internal physical boundaries, $\Gamma^{\mathrm{D,A}}$ and $\Gamma^{\mathrm{D,B}}$, respectively, corresponding to circumferences centered at point $(0,0)$ and with radius $r_{\mathrm{E}} = 1$ and $r_{\mathrm{I}} = 0.5$, respectively (see Figure 3.8). Physical subdomains Ω^{A} and Ω^{B} correspond to the outer and inner layers separated with physical interface Γ^{C} that corresponds to a circumference centered at point $(0,0)$ and with radius $r_{\mathrm{M}} = 0.75$.

To perform the simulations, successive finer uniform Delaunay triangular meshes are generated for physical subdomains Ω^{A} and Ω^{B}, where the vertices of the boundary edges coincide with the physical boundaries or interfaces (see Figure 3.8). Notice that all the cells in both meshes are polygonal and the maximum gap between the physical and the computational boundaries has magnitude order $\mathcal{O}(h^2)$, with h the characteristic mesh size.

The analytic solutions for the test case in polar coordinates (r,θ), with $r^2 = x^2 + y^2$ and



(a) Annular domain.

(b) Uniform Delaunay triangular mesh.

Figure 3.8: Domain and associated coarse mesh for the circular interface with the continuity interface conditions test case.

$\theta = \arctan(y/x)$, are given as

$$\phi^S(r,\theta) = \left(a^S \ln(r) + b^S\right)\cos\left(n^S\theta\right), \quad \text{in } \Omega^S, \tag{3.65}$$

where $n^S \in \mathbb{R}$ are given parameters and $a^S, b^S \in \mathbb{R}$ are parameters to determine. Notice that, although the analytic solutions are provided in polar coordinates, the problem is numerically solved in Cartesian coordinates. Dirichlet boundary conditions are prescribed on both physical boundaries, where for external physical boundary $\Gamma^{\mathrm{D,A}}$ the associated boundary condition function is periodic and given as $g^{\mathrm{D,A}}(r = r_{\mathrm{E}}, \theta) = \cos\left(n^{\mathrm{A}}\theta\right)$, while for internal physical boundary $\Gamma^{\mathrm{D,B}}$ the associated boundary condition function is constant and given as $g^{\mathrm{D,B}}(r = r_{\mathrm{I}}, \theta) = 0$. On the physical interface, the continuity interface conditions are prescribed and parameters a^{A}, a^{B}, b^{A}, and b^{B} in the analytic solutions are determined such that the Dirichlet boundary conditions and the continuity interface conditions are simultaneously satisfied. Constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x})$ and $\kappa^{\mathrm{B}}(\boldsymbol{x})$ are addressed, for which the parameters are determined as

$$a^{\mathrm{A}} = -c\kappa^{\mathrm{B}}, \tag{3.66}$$

$$a^{\mathrm{B}} = -c\kappa^{\mathrm{A}}, \tag{3.67}$$

$$b^{\mathrm{A}} = c\left(\kappa^{\mathrm{A}} \ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + \kappa^{\mathrm{B}} \ln(r_{\mathrm{M}})\right), \tag{3.68}$$

$$b^{\mathrm{B}} = c\kappa^{\mathrm{A}} \ln(r_{\mathrm{I}}), \tag{3.69}$$

$$c = \frac{1}{\kappa^{\mathrm{A}} \ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + \kappa^{\mathrm{B}} \ln\left(\frac{r_{\mathrm{M}}}{r_{\mathrm{E}}}\right)}. \tag{3.70}$$

On the other side, the velocities in both physical subdomains are circular and given in polar coordinates as

$$\boldsymbol{u}^S(r,\theta) = \omega^S r \hat{\boldsymbol{\theta}}, \quad \text{in } \Omega^S, \tag{3.71}$$

with given parameters $\omega^S \in \mathbb{R}$. Vectors $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{\theta}}$ are the orthogonal unit vectors in the directions of increasing r and θ, respectively. Notice that, with the above velocities, no convection occurs through the physical boundaries and interface.

The associated source term functions are obtained after substituting analytic solutions (3.65) into Equation (3.1), given in polar coordinates as

$$f^S(r,\theta) = \frac{\left(\kappa^S \left(n^S\right)^2 \cos\left(n^S\theta\right)\left(a^S \ln(r) + b^S\right)\right)}{r^2} - n^S \omega^S \sin\left(n^S\theta\right)\left(a^S \ln(r) + b^S\right), \quad \text{in } \Omega^S. \tag{3.72}$$

Two cases are addressed, namely a low thermal conductivity ratio case with constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 2$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 1$ and a high thermal conductivity ratio case with constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 100$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 1$. In both cases, the parameters in the analytic solutions are $n^{\mathrm{A}} = n^{\mathrm{B}} = 4$ and the velocity functions in each subdomain have opposite directions with parameters $\omega^{\mathrm{A}} = 1$ and $\omega^{\mathrm{B}} = -1$, as shown in Figures 3.9. The resulting analytic solutions for the low and high thermal conductivity ratio cases are shown in Figures 3.10 and the associated source terms are shown in Figures 3.11. Notice that, although the analytic solutions are continuous on the physical interface, the corresponding normal derivatives and source terms are discontinuous due to the discontinuous thermal conductivity functions in each physical subdomain.

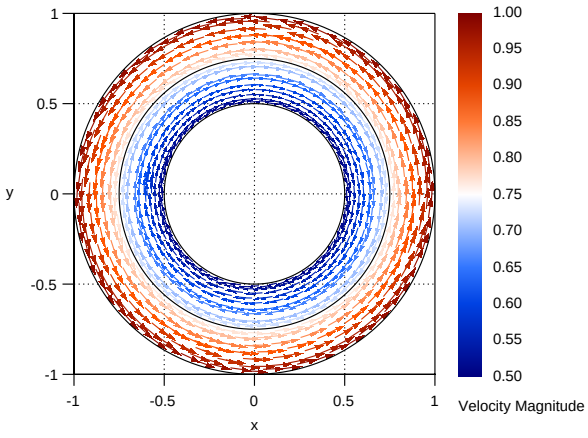
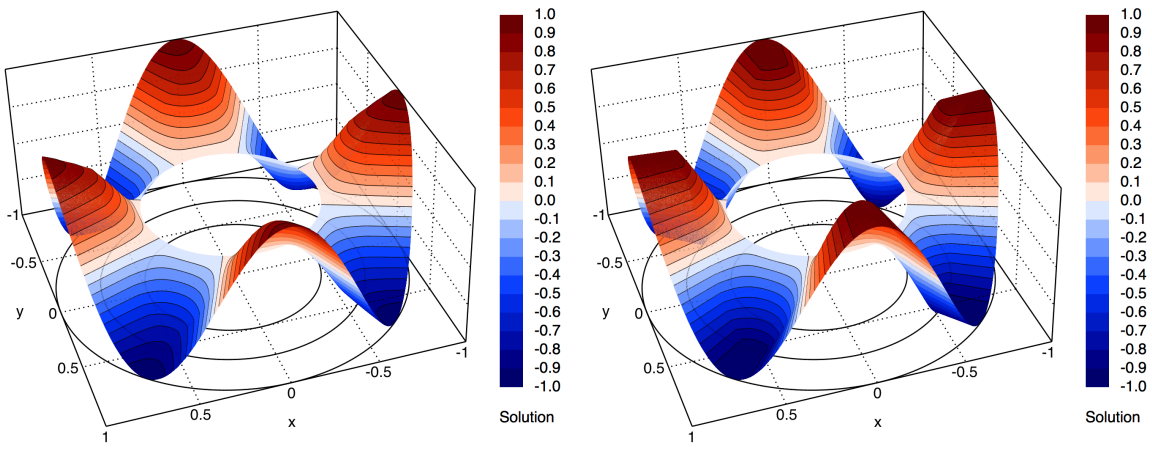


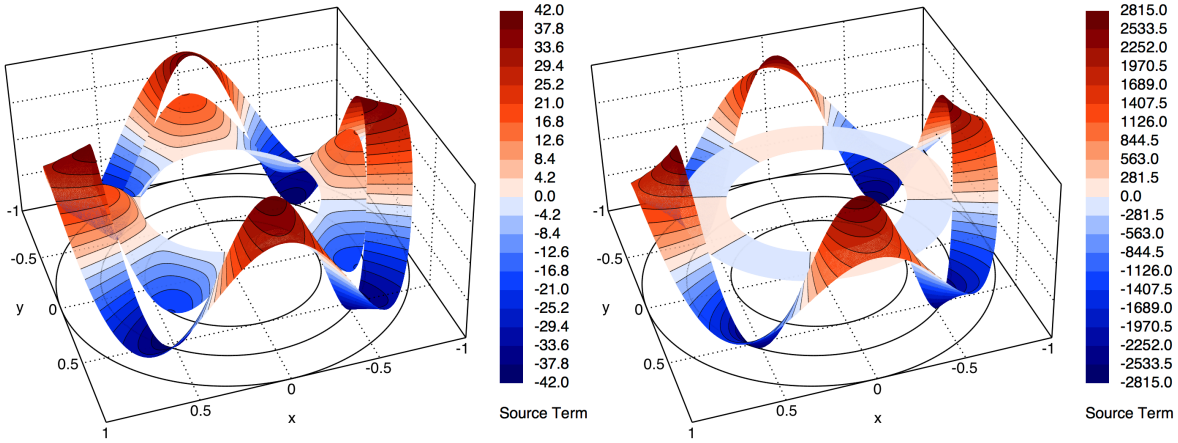
Figure 3.9: Velocities for the circular interface with the continuity interface conditions test case.



(a) Low thermal conductivity ratio case.

(b) High thermal conductivity ratio case.

Figure 3.10: Analytic solutions for the circular interface with the continuity interface conditions test case.



(a) Low thermal conductivity ratio case.

(b) High thermal conductivity ratio case.

Figure 3.11: Source terms for the circular interface with the continuity interface conditions test case.

3.6.1.1 Low thermal conductivity ratio case

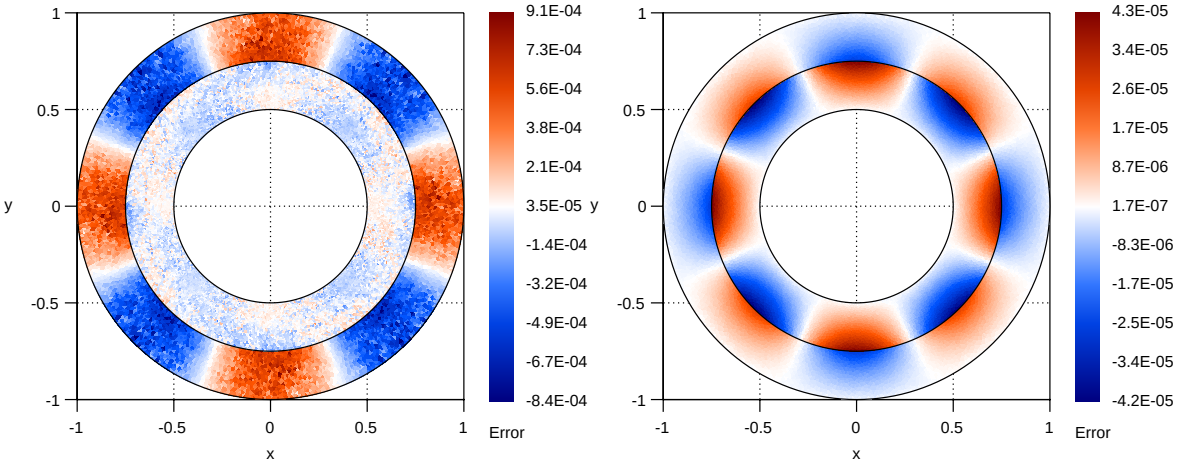
The low thermal conductivity ratio case is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, configurations ND, and Methods 1 to 5 are carried out. The measured relative errors and convergence orders are reported in Table 3.3 and the relative error distribution is shown in Figures 3.12 to 3.16 for a mesh consisting of 18 258 cells. Notice that configuration ND prescribes the Dirichlet boundary condition in subproblem P^{A} (with the highest thermal conductivity), while the Neumann boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity).

- The results from Method 1 confirm that the method for polygonal domains provides a second-order of convergence regardless of the polynomial degree, although more accurate results (with ratios up to 25) are obtained with polynomial degrees $d = 3, 5$ than with polynomial

Table 3.3: Relative errors and convergence orders obtained in the circular interface with the continuity interface conditions test case with a low thermal conductivity ratio and uniform Delaunay triangular meshes.

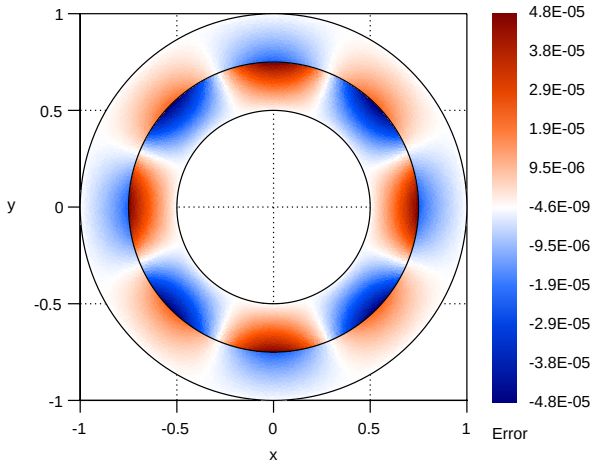
	$d = 1$				$d = 3$				$d = 5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Method 1												
1 052	3.51E−03	–	1.03E−02	–	1.47E−04	–	4.80E−04	–	1.65E−04	–	6.31E−04	–
4 484	8.50E−04	1.96	3.15E−03	1.64	3.66E−05	1.92	1.56E−04	1.55	3.69E−05	2.06	1.82E−04	1.72
18 258	2.32E−04	1.85	9.09E−04	1.77	9.67E−06	1.90	4.27E−05	1.85	9.49E−06	1.93	4.77E−05	1.90
73 834	5.79E−05	1.99	2.70E−04	1.74	2.39E−06	2.00	1.16E−05	1.87	2.37E−06	1.98	1.21E−05	1.96
292 928	1.54E−05	1.92	7.79E−05	1.80	5.95E−07	2.02	2.98E−06	1.97	5.93E−07	2.01	3.05E−06	2.00
Method 2												
1 052	3.65E−03	–	1.07E−02	–	1.98E−04	–	7.40E−04	–	1.94E−04	–	6.75E−04	–
4 484	8.65E−04	1.99	3.22E−03	1.65	4.77E−05	1.96	1.64E−04	2.08	4.84E−05	1.92	1.69E−04	1.91
18 258	2.37E−04	1.85	9.20E−04	1.79	1.24E−05	1.91	4.38E−05	1.88	1.25E−05	1.93	4.41E−05	1.91
73 834	5.83E−05	2.00	2.75E−04	1.73	3.12E−06	1.98	1.11E−05	1.96	3.12E−06	1.98	1.11E−05	1.97
292 928	1.55E−05	1.92	7.90E−05	1.81	7.80E−07	2.01	2.80E−06	2.00	7.80E−07	2.01	2.80E−06	2.00
Method 3												
1 052	3.65E−03	–	1.07E−02	–	7.75E−05	–	4.06E−04	–	3.85E−05	–	1.91E−04	–
4 484	8.65E−04	1.99	3.22E−03	1.65	9.27E−06	2.93	4.03E−05	3.19	4.26E−07	6.21	2.44E−06	6.02
18 258	2.37E−04	1.85	9.20E−04	1.79	1.44E−06	2.65	6.17E−06	2.67	1.53E−08	4.74	7.68E−08	4.93
73 834	5.83E−05	2.00	2.75E−04	1.73	1.55E−07	3.19	6.82E−07	3.15	3.37E−10	5.47	1.86E−09	5.33
292 928	1.55E−05	1.92	7.90E−05	1.81	1.97E−08	2.99	7.89E−08	3.13	1.15E−11	4.90	5.22E−11	5.18
Method 4												
1 052	3.66E−03	–	1.08E−02	–	5.79E−05	–	2.70E−04	–	2.08E−05	–	1.16E−04	–
4 484	8.71E−04	1.98	3.26E−03	1.65	3.48E−06	3.88	2.12E−05	3.50	3.29E−07	5.72	1.96E−06	5.63
18 258	2.38E−04	1.85	9.24E−04	1.80	4.80E−07	2.82	2.65E−06	2.96	8.46E−09	5.22	4.18E−08	5.48
73 834	5.88E−05	2.00	2.78E−04	1.72	4.86E−08	3.28	2.72E−07	3.26	1.90E−10	5.43	9.69E−10	5.39
292 928	1.56E−05	1.92	7.95E−05	1.82	6.52E−09	2.92	3.15E−08	3.13	6.07E−12	5.00	4.75E−11	4.37
Method 5												
1 052	3.36E−03	–	9.96E−03	–	6.89E−05	–	2.46E−04	–	2.19E−05	–	1.18E−04	–
4 484	8.45E−04	1.90	3.03E−03	1.64	3.82E−06	3.99	2.06E−05	3.42	2.62E−07	6.11	1.82E−06	5.75
18 258	2.28E−04	1.86	8.93E−04	1.74	2.70E−07	3.78	1.43E−06	3.80	3.53E−09	6.13	2.47E−08	6.13
73 834	5.80E−05	1.96	2.63E−04	1.75	1.47E−08	4.16	9.72E−08	3.85	4.37E−11	6.29	4.82E−10	5.63
292 928	1.55E−05	1.91	7.62E−05	1.80	9.61E−10	3.96	8.72E−09	3.50	4.29E−12	–	2.98E−11	–

degree $d = 1$. Observing the relative error distribution shown in Figures 3.12, for a uniform Delaunay triangular mesh with 18 258 cells, the accuracy is noticeably deteriorated in the vicinity of the physical interface with polynomial degrees $d = 3,5$ in contrast to polynomial degree $d = 1$. Moreover, for the same polynomial degrees, smaller errors are observed in the vicinity of the external and internal physical boundaries, which is the expected behaviour since the prescribed Dirichlet boundary conditions are correctly fulfilled with the use of the ROD method.



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.

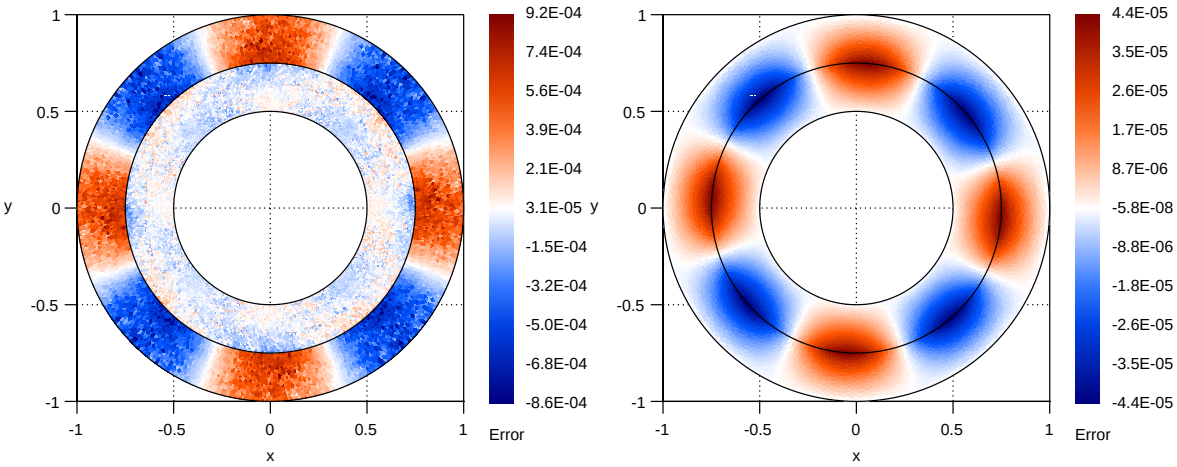


(c) Polynomial degree $d = 5$.

Figure 3.12: Relative error distribution obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio, method 1, and configuration ND.

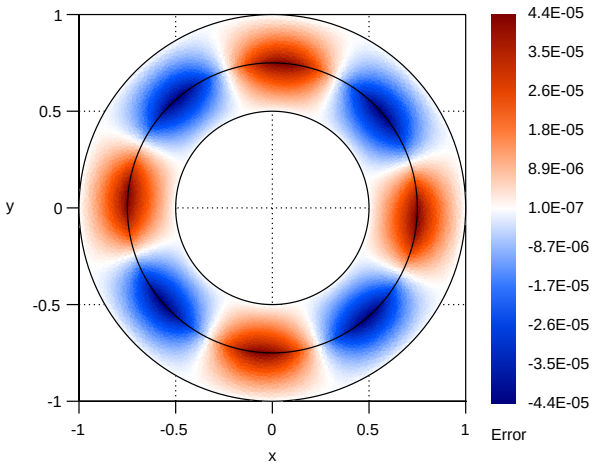
- In Method 2, no improvements in terms of accuracy or convergence orders are observed in comparison to Method 1, which is also confirmed from the observation of the relative error distribution shown in Figure 3.13, for a uniform Delaunay triangular mesh with 18 258 cells. A discretization scheme with symmetric numerical conductive heat fluxes associated with the boundary edges preserves the conservation of the conductive heat flux on the straight computational boundaries rather than on the curved physical boundaries, corresponding to the

physical interfaces. Although the temperature continuity is fulfilled on the physical boundaries with the use of the ROD method, the geometrical mismatch results in a relative error with the same magnitude order as the expected gap for the conservation of the conductive heat flux. Moreover, the approximate solution accuracy obtained with Method 2 is worse than with Method 1. Indeed, it seems worst to fulfil the temperature continuity on the curved physical boundaries and the conductive heat flux conservation on the straight computational boundaries, rather than fulfilling both conditions on the straight computational boundaries.



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.

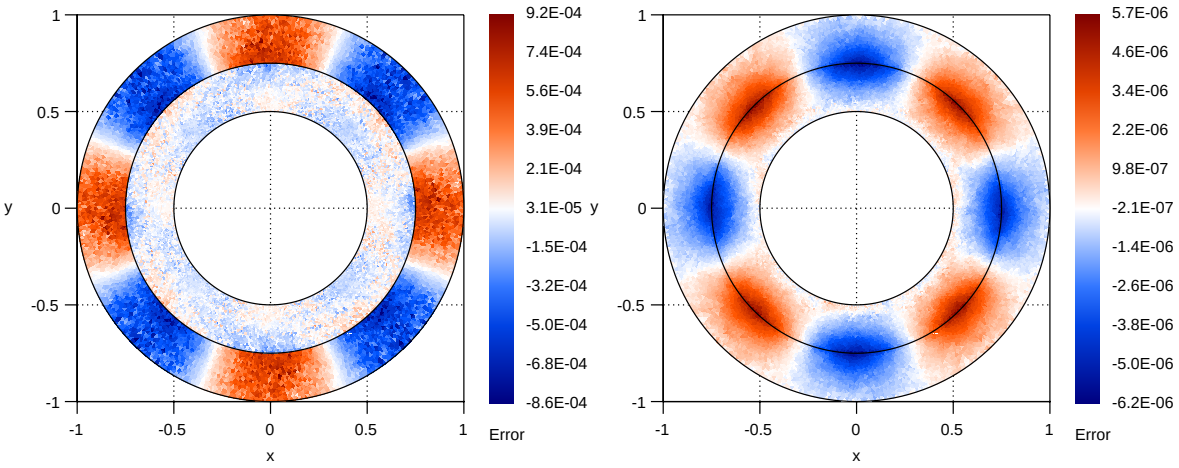


(c) Polynomial degree $d = 5$.

Figure 3.13: Relative error distribution obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio, method 2, and configuration ND.

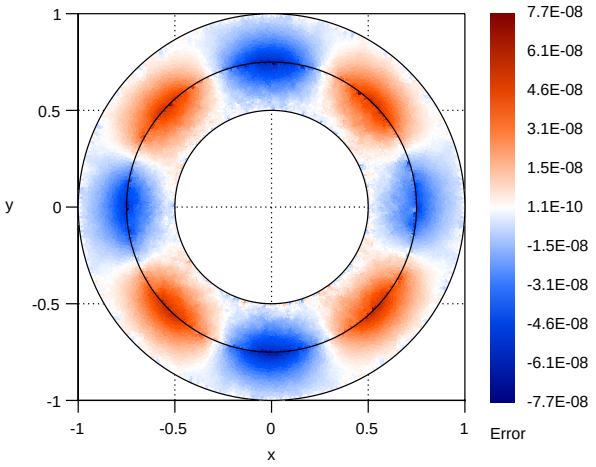
- The second-order of convergence observed in Methods 1 and 2 with polynomial degrees $d = 3, 5$ is overcame in Method 3 and the results prove that the ROD method succeeds in adequately fulfilling the prescribed boundary conditions on the physical boundaries with the use of polygonal meshes. Besides the use of the ROD method, the proposed discretization scheme for curved domains is also needed to compute the numerical convective and conductive heat fluxes on the boundary edges, avoiding the issues observed in Method 2. However, although a

very high-order of convergence is provided, the optimal fourth- and sixth-orders of convergence with polynomial degrees $d = 3, 5$, respectively, are not achieved. Moreover, the observation of the relative error distribution shown in Figure 3.14, for a uniform Delaunay triangular mesh with 18 258 cells, reveals that the accuracy deteriorates in the vicinity of the physical interface, which limits the convergence order of the proposed method to the sub-optimal.



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.

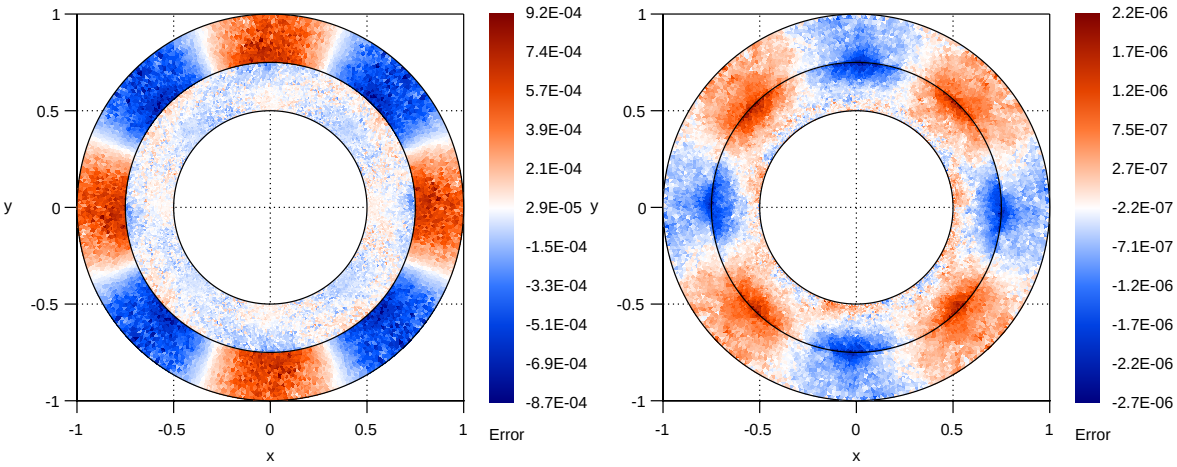


(c) Polynomial degree $d = 5$.

Figure 3.14: Relative error distribution obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio, method 3, and configuration ND.

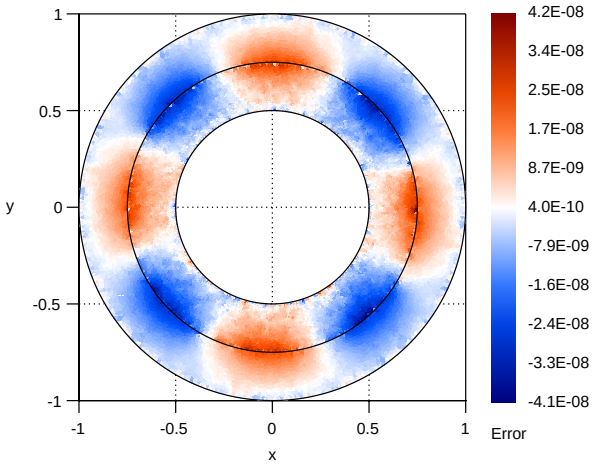
- In Method 4, the accuracy of the approximate solution is improved, when compared with Method 3, but the convergence remains limited to the third- and fifth-orders with polynomial degrees $d = 3, 5$, respectively. Moreover, the relative error distribution shown in Figure 3.15, for a uniform Delaunay triangular mesh with 18 258 cells, reveals, once again, that the accuracy deteriorates in the vicinity of the physical interface, becoming more evident for higher polynomial degrees.

- In Method 5, the optimal fourth- and sixth-orders of convergence are effectively achieved with



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



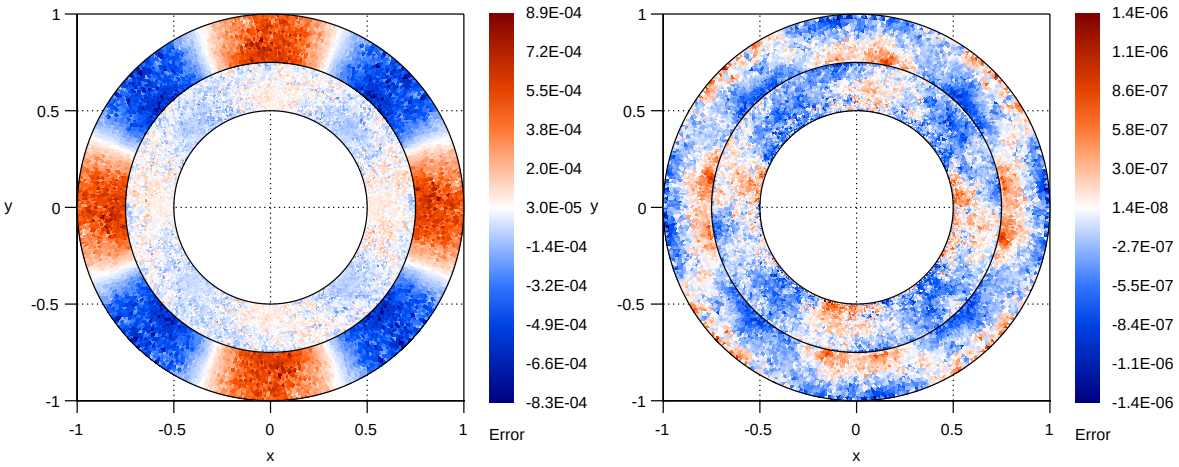
(c) Polynomial degree $d = 5$.

Figure 3.15: Relative error distribution obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio, method 4, and configuration ND.

polynomial degrees $d = 3, 5$, respectively, both in terms of the L^1- and L^∞-norms. No localized accuracy deterioration is observed in the relative error distribution shown in Figure 3.16, for a uniform Delaunay triangular mesh with 18 258 cells, which is more reasonable and uniform than with the previous methods. Notice that, with polynomial degree $d = 5$, the convergence order for the last mesh is not provided since the measured relative errors in the L^1- and L^∞-norms correspond to the maximum numerical precision in the simulations (machine precision multiplied by condition numbers).

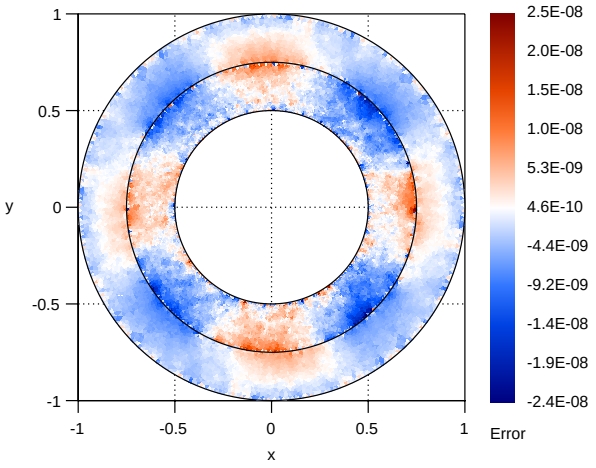
3.6.1.2 High thermal conductivity ratio case

The high thermal conductivity ratio case is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, configurations DN and ND, and Method 5 are carried out. The measured relative errors and convergence orders are reported in



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



(c) Polynomial degree $d = 5$.

Figure 3.16: Relative error distribution obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio, method 5, and configuration ND.

Table 3.4 and the relative error distribution is shown in Figures 3.17 and 3.18, for a uniform Delaunay triangular mesh with 18 258 cells. Notice that configuration DN prescribes the Neumann boundary condition in subproblem P^{A} (with the highest thermal conductivity). In contrast, the Dirichlet boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity) and configuration ND prescribes the opposite situation.

The optimal convergence orders are effectively achieved with polynomial degrees $d = 1, 3, 5$, both in terms of the L^1- and L^∞-norms, with both partitioning configurations. However, configuration ND provides approximate solutions that are slightly more accurate than configuration DN, both in terms of the L^1- and L^∞-norms. For configuration ND, the magnitude of the relative errors in both subdomains is comparable in the vicinity of the physical interface with polynomial degrees $d = 3, 5$. On the other side, for configuration DN, subdomain Ω^{B} conatains larger errors, both in terms of the L^1- and L^∞-norms, than subdomain Ω^{A}, which is in accordance with the results reported in Table 3.4. Although the optimal convergence orders are effectively achieved with both partitioning configurations,

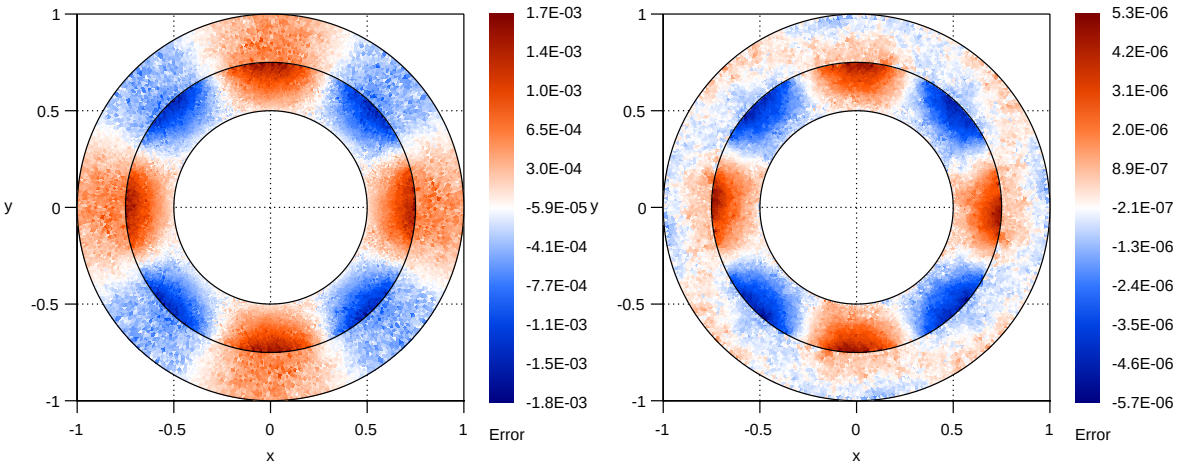
the results confirm that small variations in terms of accuracy are obtained. The results in Costa et al., 2018 [65], for the treatment of convection-diffusion problems with curved physical boundaries, support one possible explanation for this behaviour. The authors reported that, for the same analytic solution, prescribing Neumann boundary conditions in the problem usually leads to larger errors, both in terms of the L^1- and L^∞-norms, than with Dirichlet boundary conditions prescribed on the same physical boundary. Consequently, it suggests that prescribing the Neumann boundary condition in the subproblem with the smoothest analytic solution and the Dirichlet boundary condition in the other provides better accuracy. Indeed, the Neumann boundary condition prescribed in subproblem P^B and the Dirichlet boundary condition prescribed in subproblem P^A corresponds to configuration ND. However, many factors and interactions exist in these simulations and, therefore, different situations can lead to other behaviours.

Table 3.4: Relative errors and convergence orders obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio and uniform Delaunay triangular meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Configuration DN												
1 052	6.09E−03	–	2.45E−02	–	3.00E−04	–	1.86E−03	–	5.91E−05	–	3.67E−04	–
4 484	1.56E−03	1.88	6.98E−03	1.73	1.60E−05	4.04	8.34E−05	4.29	7.36E−07	6.05	3.60E−06	6.38
18 258	4.22E−04	1.86	1.83E−03	1.90	9.78E−07	3.98	5.67E−06	3.83	1.37E−08	5.68	1.01E−07	5.09
73 834	1.03E−04	2.02	5.08E−04	1.84	5.85E−08	4.03	7.45E−07	2.91	1.81E−10	6.19	1.86E−09	5.72
292 928	2.92E−05	1.83	1.43E−04	1.84	3.11E−09	4.26	2.37E−08	5.01	7.58E−12	–	9.30E−11	–
Configuration ND												
1 052	3.61E−03	–	1.03E−02	–	1.15E−04	–	4.24E−04	–	4.08E−05	–	2.18E−04	–
4 484	8.75E−04	1.95	3.42E−03	1.52	9.89E−06	3.38	4.18E−05	3.20	6.74E−07	5.66	3.81E−06	5.58
18 258	2.35E−04	1.87	9.83E−04	1.77	5.35E−07	4.16	2.54E−06	3.99	9.99E−09	6.00	7.18E−08	5.66
73 834	5.90E−05	1.98	2.94E−04	1.73	3.50E−08	3.90	2.74E−07	3.18	1.56E−10	5.95	1.06E−09	6.03
292 928	1.64E−05	1.85	1.00E−04	1.56	2.12E−09	4.07	1.28E−08	4.44	4.10E−12	–	2.61E−11	–

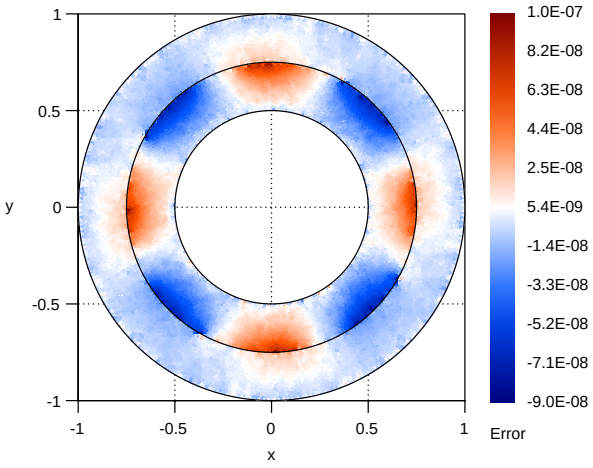
3.6.2 Rose-shaped interface with the continuity interface conditions test case

In this test case, the conjugate heat transfer problem is addressed with a complex curved physical interface prescribed with the continuity interface conditions. An annular domain Ω is considered consisting of an external and internal physical boundaries, $\Gamma^{\mathrm{D,A}}$ and $\Gamma^{\mathrm{D,B}}$, respectively, corresponding to circumferences centered at point $(0,0)$ and with radius $r_\mathrm{E}=1$ and $r_\mathrm{I}=0.5$, respectively (see Figure 3.19). Physical subdomains Ω^A and Ω^B correspond to the outer and inner layers separated with physical interface Γ^C that corresponds to the diffeomorphic transformation applied to a circumference centered at point $(0,0)$ and with radius $r_\mathrm{M}=0.75$. The diffeomorphic transformation consists in a periodic mapping from the circumference to the produced curve, given in



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



(c) Polynomial degree $d = 5$.

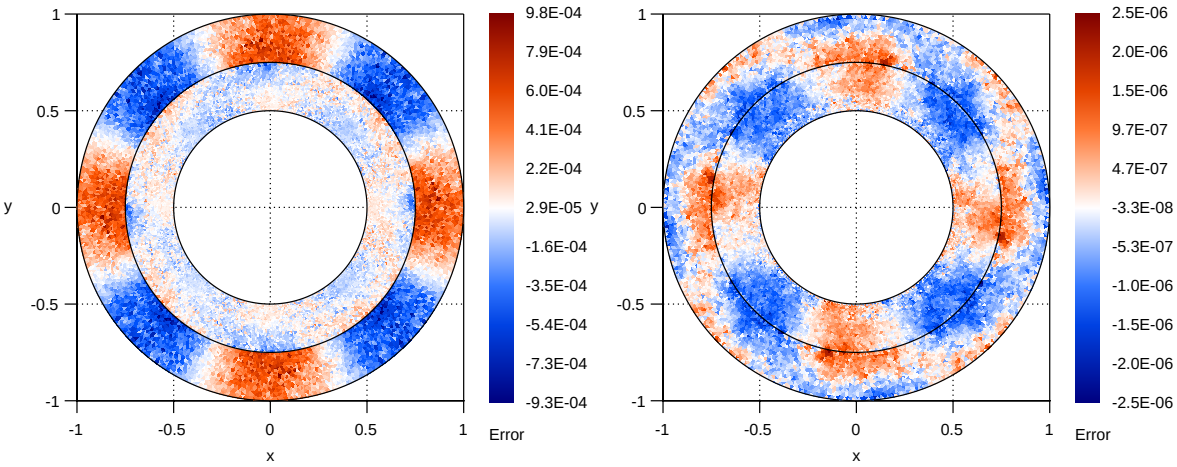
Figure 3.17: Relative error distribution obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio and configuration DN.

polar coordinates as

$$\begin{bmatrix} r \\ \theta \end{bmatrix} \rightarrow \begin{bmatrix} T(\theta) \\ \theta \end{bmatrix} = \begin{bmatrix} r_{\mathrm{M}}(1+\beta_\theta \cos(\alpha_\theta \theta)) \\ \theta \end{bmatrix}, \quad \text{on } \Gamma^{\mathrm{C}}, \tag{3.73}$$

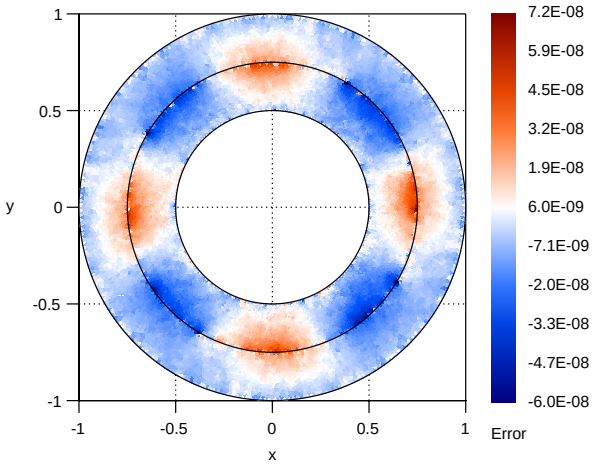
where function $T(\theta)$ is a parameterization in terms of θ with parameters $r_{\mathrm{M}}, \alpha_\theta, \beta_\theta \in \mathbb{R}$ and corresponds to the varying ray of the produced curve. Because it resembles rose petals, the produced curve is referred to as the rose-shaped interface. Notice that the circular interface is recovered for parameter $\alpha_\theta = 0$.

Unit normal vector function on the physical interface, denoted as $\boldsymbol{n}^{\mathrm{A}} := \boldsymbol{n}^{\mathrm{A}}(\theta)$ (notice that $\boldsymbol{n}^{\mathrm{B}} = -\boldsymbol{n}^{\mathrm{A}}$), from physical subdomain Ω^{A} to physical subdomain Ω^{B}, is obtained applying the chain



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



(c) Polynomial degree $d = 5$.

Figure 3.18: Relative error distribution obtained in the circular interface with the continuity interface conditions test case with a high thermal conductivity ratio and configuration ND.

rule to function $T(\theta)$ and is given in polar coordinates as

$$\boldsymbol{n}^{\mathrm{A}}(\theta) = \frac{1}{\sqrt{(T(\theta))^2 + (\partial_\theta T(\theta))^2}} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix} \begin{bmatrix} T(\theta) \\ \partial_\theta T(\theta) \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}} & \hat{\boldsymbol{y}} \end{bmatrix}, \quad \text{on } \Gamma^{\mathrm{C}}, \tag{3.74}$$

where $\partial_\theta T(\theta) = r_{\mathrm{M}}(-\alpha_\theta \beta_\theta \sin(\alpha_\theta \theta))$. Vectors $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are the orthogonal unit vectors in the directions of increasing x and y, respectively.

The global mapping from domain Ω with the rose-shaped interface to an annular domain with a circular interface, denoted Ω', is provided with vector function $\boldsymbol{T}_\Omega(r, \theta)$ given in polar coordinates as

$$\boldsymbol{T}_{\Omega \to \Omega'}(r, \theta) = \begin{bmatrix} D(r, \theta) \\ \theta \end{bmatrix}, \quad D(r, \theta) = d_1(\theta) r^2 + d_2(\theta) r + d_3(\theta), \quad \text{in } \Omega, \tag{3.75}$$

where the result is provided in polar coordinates. Functions $d_1(\theta)$, $d_2(\theta)$, and $d_3(\theta)$ are sought such that the annular domain with a circular interface is recovered, that is, $D(r_{\mathrm{I}},\theta) = r_{\mathrm{I}}$, $D(r_{\mathrm{E}},\theta) = r_{\mathrm{E}}$, and $D(T(\theta),\theta) = r_{\mathrm{M}}$, and are determined as

$$d_1(\theta) = -c\beta_\theta \cos(\alpha_\theta \theta), \tag{3.76}$$

$$d_2(\theta) = 1 + c(r_{\mathrm{I}} + r_{\mathrm{E}})\beta_\theta \cos(\alpha_\theta \theta), \tag{3.77}$$

$$d_3(\theta) = -c\beta_\theta r_{\mathrm{I}} r_{\mathrm{E}} \cos(\alpha_\theta \theta), \tag{3.78}$$

$$c = \frac{1}{(r_{\mathrm{M}} - r_{\mathrm{I}} + \beta_\theta \cos(\alpha_\theta \theta))(r_{\mathrm{M}} - r_{\mathrm{E}} + \beta_\theta \cos(\alpha_\theta \theta))}. \tag{3.79}$$

To perform the simulations, successive finer uniform Delaunay triangular or uniform structured quadrilateral meshes are generated for physical subdomains Ω^{A} and Ω^{B} (see Figure 3.19). As for the previous test case, the vertices of the boundary edges coincide with the physical boundaries and interface and all the cells in both meshes are polygonal such that the maximum gap between the physical and the computational boundaries and interface has magnitude order $\mathcal{O}(h^2)$, with h the characteristic mesh size.

The analytic solutions for this test case are given in polar coordinates as

$$\phi^S(r,\theta) = a^S + b^S \ln(D(r,\theta)), \quad \text{in } \Omega^S, \tag{3.80}$$

where $a^S, b^S \in \mathbb{R}$ are parameters to determine. Notice that, although the analytic solutions are provided in polar coordinates, the problem is numerically solved in Cartesian coordinates. Dirichlet boundary conditions are prescribed on both external and internal boundaries, $\Gamma^{\mathrm{D,A}}$ and $\Gamma^{\mathrm{D,B}}$, with constant boundary condition functions given as $g^{\mathrm{D,A}}(r = r_{\mathrm{I}},\theta) = 1$ and $g^{\mathrm{D,B}}(r = r_{\mathrm{E}}) = 0$, respectively. On the interface, the continuity interface conditions are prescribed and parameters a^{A}, a^{B}, b^{A}, and b^{B} are determined such that the Dirichlet boundary conditions and the continuity interface conditions are simultaneously satisfied. Constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x})$ and $\kappa^{\mathrm{B}}(\boldsymbol{x})$ are addressed, for which the parameters are determined as

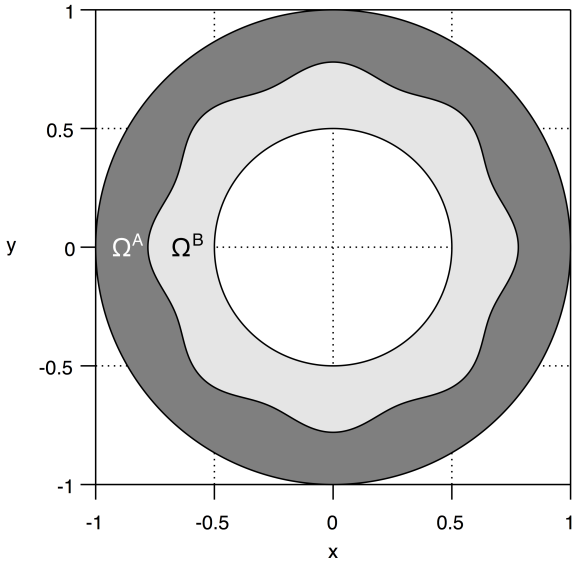
$$a^{\mathrm{A}} = -c\kappa^{\mathrm{B}}, \tag{3.81}$$

$$a^{\mathrm{B}} = -c\kappa^{\mathrm{A}}, \tag{3.82}$$

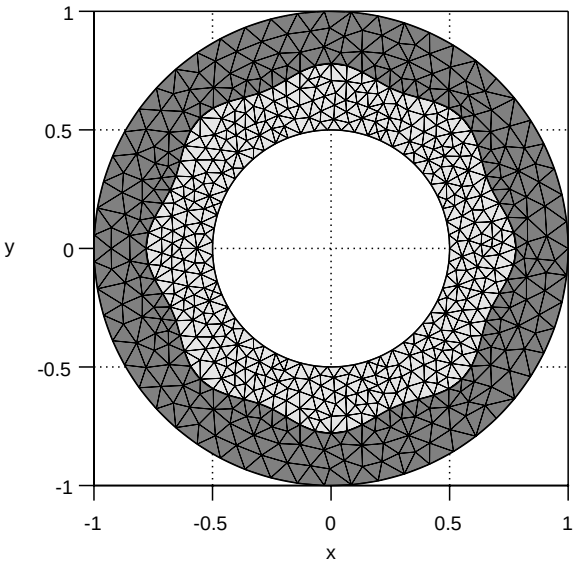
$$b^{\mathrm{A}} = c\left(\kappa^{\mathrm{A}} \ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + \kappa^{\mathrm{B}} \ln(r_{\mathrm{M}})\right), \tag{3.83}$$

$$b^{\mathrm{B}} = c\kappa^{\mathrm{A}} \ln(r_{\mathrm{I}}), \tag{3.84}$$

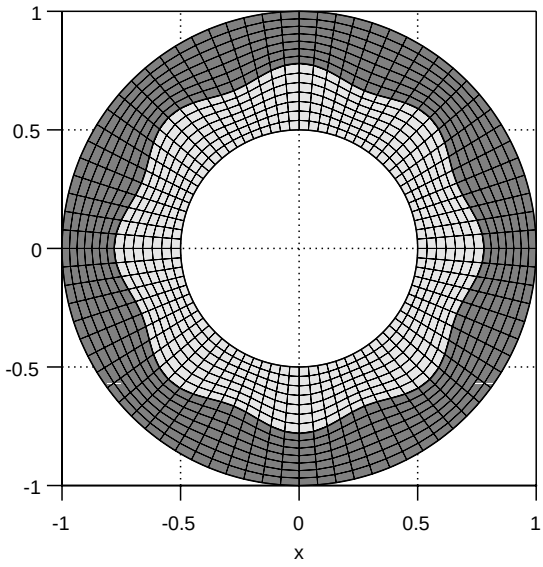
$$c = \frac{1}{\kappa^{\mathrm{A}} \ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + \kappa^{\mathrm{B}} \ln(r_{\mathrm{M}} r_{\mathrm{E}})}. \tag{3.85}$$



(a) Annular domain.



(b) Uniform Denaulay triangular meshes.



(c) Uniform structured quadrilateral mesh.

Figure 3.19: Domain and associated coarse meshes for the rose-shaped interface with the continuity interface conditions test case.

The velocities for both subdomains are circular on the internal and external physical boundaries and follows the rose-shaped curve on the physical interface, given in polar coordinates as

$$\boldsymbol{u}^S(r,\theta) = \omega^S r\left(\frac{(r-r_0)\partial_\theta T(\theta)}{T(\theta)-r_0}\hat{\boldsymbol{r}} + \hat{\boldsymbol{\theta}}\right), \quad \text{in } \Omega^S, \tag{3.86}$$

where parameters $\omega^{\mathrm{A}}, \omega^{\mathrm{B}} \in \mathbb{R}$ and $r_0 = r_{\mathrm{E}}$ for physical subdomains Ω^{A} and $r_0 = r_{\mathrm{I}}$ for physical subdomain Ω^{B}. Notice that no convection occurs through the boundaries or the interface.

The associated source term functions are obtained substituting analytic solutions (3.80) into

Equation (3.1), resulting in lengthy and complex expressions, which are not provided for the sake of compactness.

As previously, two cases are addressed, namely a low thermal conductivity ratio case with constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 2$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 1$ and a high thermal conductivity ratio case with constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 100$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 1$. In both cases, the velocities in each physical subdomain have opposite directions assigning parameters $\omega^{\mathrm{A}} = 1$ and $\omega^{\mathrm{B}} = -1$, as shown in Figures 3.20. Parameters $\alpha_\theta = 8$ and $\beta_\theta = 0.04$ are considered resulting in a diffeomorphic transformation. The resulting analytic solutions for the low and the high thermal conductivity ratio cases are shown in Figures 3.21 and the associated source terms are shown in Figures 3.22. As for the previous test case, neither the derivatives normal to the physical interface of the analytic solutions nor the source terms are continuous on the physical interface, although the conservation of the conductive heat flux is preserved.

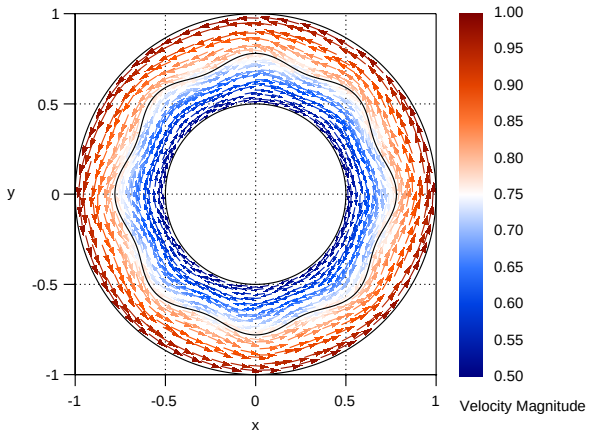
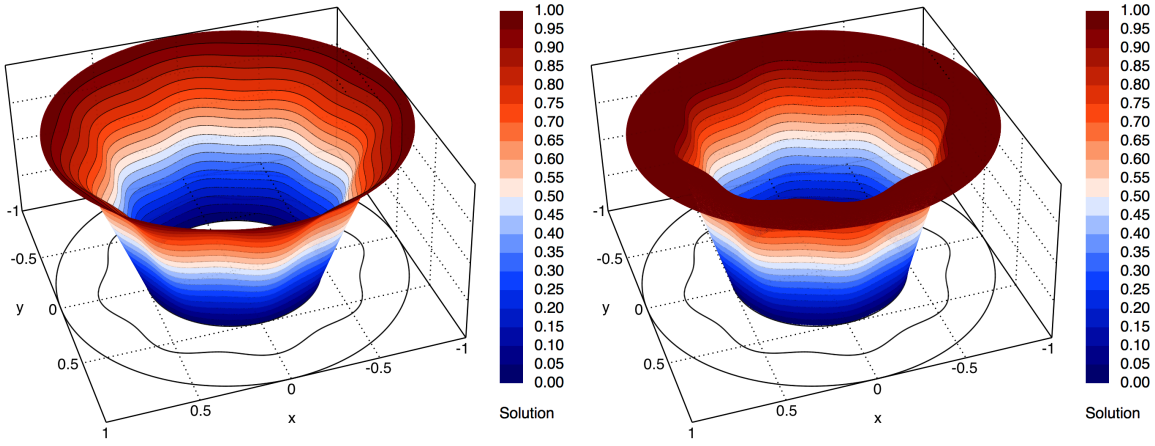


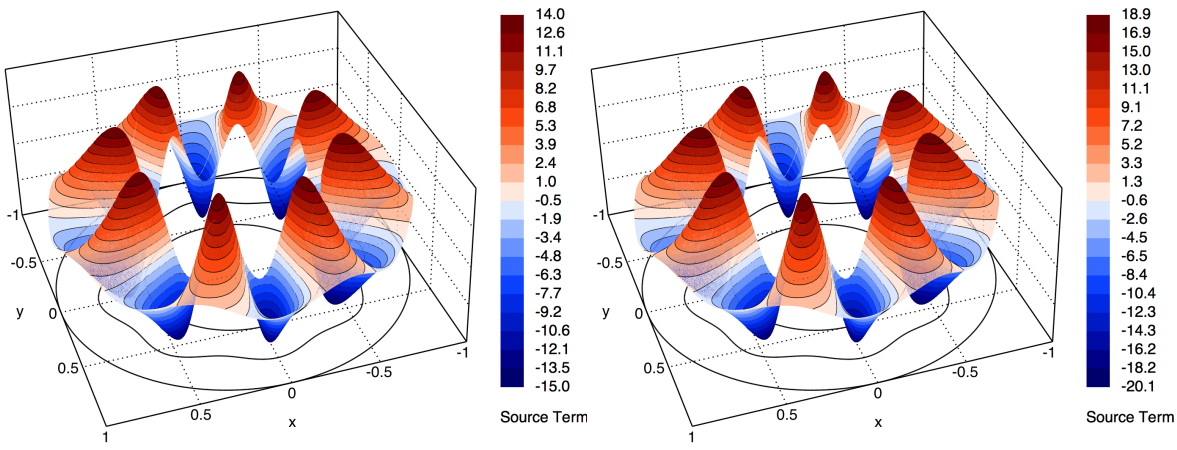
Figure 3.20: Velocities for the rose-shaped interface with the continuity interface conditions test case.



(a) Low thermal conductivity ratio case.

(b) High thermal conductivity ratio case.

Figure 3.21: Analytic solutions for the rose-shaped interface with the continuity interface conditions test case.



(a) Low thermal conductivity ratio case.

(b) High thermal conductivity ratio case.

Figure 3.22: Source terms for the rose-shaped interface with the continuity interface conditions test case.

3.6.2.1 Low thermal conductivity ratio case

The low thermal conductivity ratio case is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, configurations DN and ND, and Method 5 are carried out. The measured relative errors and convergence orders are reported in Table 3.5 and the relative error distribution is shown in Figures 3.23 and 3.24, for a uniform Delaunay triangular mesh with 18 124 cells. Notice that configuration DN prescribes the Neumann boundary condition in subproblem P^{A} (with the highest thermal conductivity). In contrast, the Dirichlet boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity) and configuration ND prescribes the opposite situation.

The optimal second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms. At the same time, an essentially non-oscillatory behaviour is observed, which supports that the proposed method is capable of handling complex curved physical interfaces with polygonal meshes. Both partitioning configurations provide comparable accuracy and convergence orders, both in terms of the L^1- and L^∞-norms. However, the relative error distribution reveals that larger errors in the vicinity of the physical interface are obtained with configuration ND and polynomial degrees $d = 3, 5$ than with configuration DN. However, the observed differences are not significant in terms of error magnitude.

The same test case is addressed with successive finer uniform structured quadrilateral meshes to verify whether the proposed method can handle other types of polygonal meshes. For that purpose, simulations with the same polynomial degrees, partitioning configurations, and method are carried out. The measured relative errors and convergence orders are reported in Table 3.6.

The optimal second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms. The results support that the proposed method is not confined to Delaunay triangular meshes and, therefore,

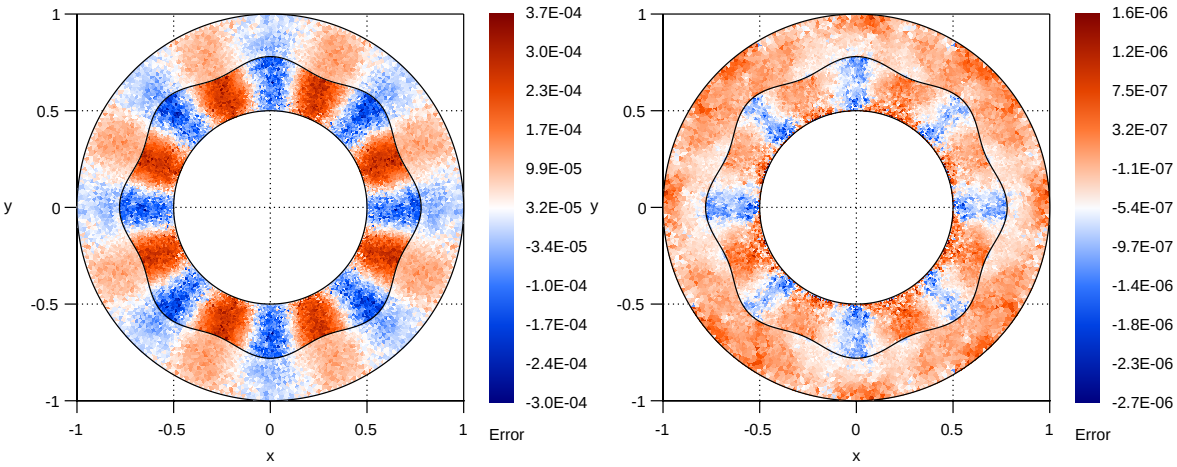
Table 3.5: Relative errors and convergence orders obtained in the rose-shaped interface with the continuity interface conditions test case with a low thermal conductivity ratio and uniform Delaunay triangular meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Configuration DN												
1 132	1.42E−03	—	5.74E−03	—	1.03E−04	—	5.32E−04	—	5.71E−05	—	2.39E−04	—
4 548	3.56E−04	1.99	1.55E−03	1.88	5.63E−06	4.19	3.76E−05	3.81	8.29E−07	6.09	6.96E−06	5.08
18 124	8.79E−05	2.02	3.67E−04	2.09	3.35E−07	4.08	2.69E−06	3.81	8.88E−09	6.56	1.03E−07	6.10
71 844	2.36E−05	1.91	1.07E−04	1.79	1.82E−08	4.23	1.89E−07	3.86	1.05E−10	6.44	2.09E−09	5.66
289 954	5.76E−06	2.02	2.62E−05	2.02	1.11E−09	4.00	1.43E−08	3.71	6.04E−12	—	6.67E−11	—
Configuration ND												
1 132	1.45E−03	—	5.54E−03	—	1.05E−04	—	6.76E−04	—	5.31E−05	—	2.38E−04	—
4 548	3.52E−04	2.03	1.47E−03	1.90	7.48E−06	3.79	3.74E−05	4.16	8.01E−07	6.03	6.99E−06	5.07
18 124	8.87E−05	1.99	3.36E−04	2.14	3.23E−07	4.55	2.66E−06	3.82	9.24E−09	6.45	1.97E−07	5.17
71 844	2.34E−05	1.93	9.62E−05	1.82	2.09E−08	3.97	1.87E−07	3.85	1.11E−10	6.43	2.09E−09	6.60
289 954	5.66E−06	2.04	2.55E−05	1.91	1.33E−09	3.95	1.42E−08	3.69	4.28E−12	—	7.04E−11	—

structured or unstructured general polygonal meshes can be used without convergence order deterioration. As in the case of uniform Delaunay triangular meshes, both partitioning configurations provide comparable accuracy and the optimal convergence orders, both in terms of the L^1- and L^∞-norms.

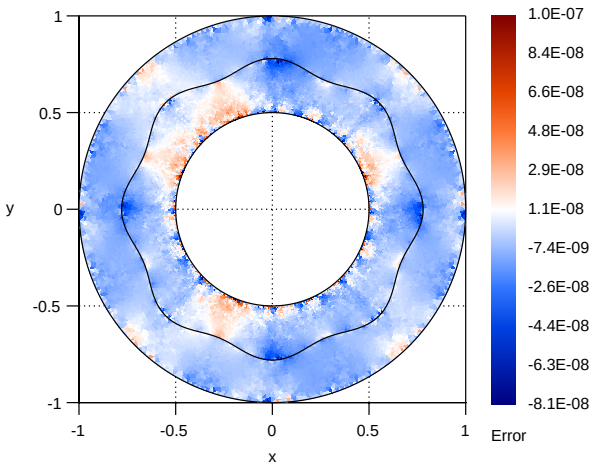
Table 3.6: Relative errors and convergence orders obtained in the rose-shaped interface with the continuity interface conditions test case with a low thermal conductivity ratio and uniform structured quadrilateral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Configuration DN												
1 120	1.41E−03	—	4.76E−03	—	1.62E−04	—	7.48E−04	—	1.76E−04	—	1.03E−03	—
4 800	3.44E−04	1.94	1.17E−03	1.92	1.09E−05	3.71	6.32E−05	3.40	3.64E−06	5.33	1.89E−05	5.50
19 840	8.54E−05	1.96	2.89E−04	1.97	7.17E−07	3.84	3.54E−06	4.06	3.51E−08	6.54	3.18E−07	5.75
80 640	2.13E−05	1.98	7.15E−05	1.99	4.78E−08	3.86	2.24E−07	3.94	7.26E−10	5.53	5.35E−09	5.83
325 120	5.31E−06	1.99	1.78E−05	1.99	5.61E−09	3.07	2.06E−08	3.42	1.02E−11	6.11	8.44E−11	5.95
Configuration ND												
1 120	1.41E−03	—	3.77E−03	—	2.49E−04	—	1.39E−03	—	1.75E−04	—	7.30E−04	—
4 800	3.46E−04	1.93	1.02E−03	1.80	1.49E−05	3.87	1.34E−04	3.21	3.28E−06	5.47	2.01E−05	4.94
19 840	8.59E−05	1.96	2.59E−04	1.93	9.05E−07	3.95	7.94E−06	3.98	3.93E−08	6.24	3.20E−07	5.84
80 640	2.14E−05	1.98	6.53E−05	1.97	6.23E−08	3.82	4.21E−07	4.19	7.76E−10	5.60	7.18E−09	5.41
325 120	5.34E−06	1.99	1.64E−05	1.98	6.28E−09	3.29	2.25E−08	4.20	1.20E−11	5.98	1.17E−10	5.90



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



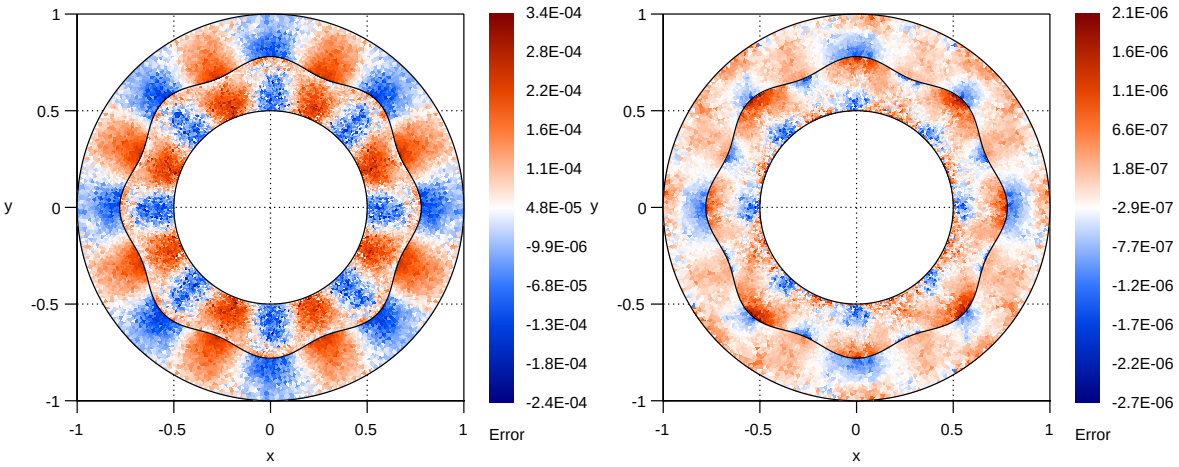
(c) Polynomial degree $d = 5$.

Figure 3.23: Relative error distribution obtained in the rose-shaped interface with the continuity interface conditions test case with a low thermal conductivity ratio and configuration DN.

3.6.2.2 High thermal conductivity ratio case

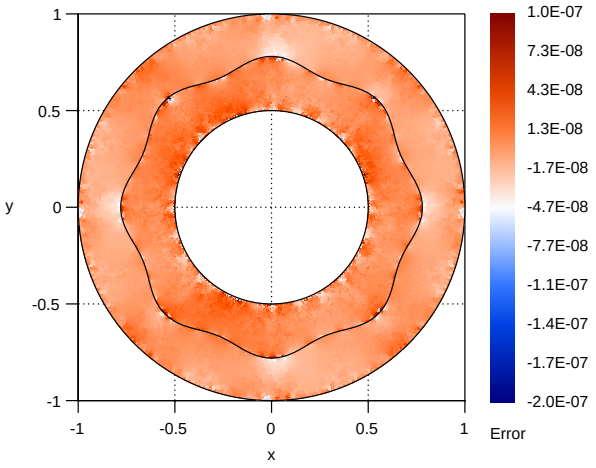
The high thermal conductivity ratio case is addressed, and simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, configurations DN and ND, and Method 5 are carried out. The measured relative errors and convergence orders are reported in Table 3.7 and the relative error distribution is shown in Figures 3.25 and 3.26, for a uniform Delaunay triangular mesh with 18 124 cells. Notice that configuration DN prescribes the Neumann boundary condition in subproblem P^{A} (with the highest thermal conductivity). In contrast, the Dirichlet boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity) and configuration ND prescribes the opposite situation.

As for the previous case, the proposed method achieves the optimal convergence orders effectively with polynomial degrees $d = 1, 3, 5$, both in terms of the L^1- and L^∞-norms. The results further support the capability of the proposed method to handle arbitrary curved physical interfaces



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



(c) Polynomial degree $d = 5$.

Figure 3.24: Relative error distribution obtained in the rose-shaped interface with the continuity interface conditions test case with a low thermal conductivity ratio and configuration ND.

prescribed with the continuity interface conditions and provide very high-order of convergence, even for large jumps of the thermal conductivity functions on the interface. Configuration DN provides a slightly better accuracy with polynomial degrees $d = 3, 5$ than configuration ND. Additionally, the relative error distribution reveals that larger errors are obtained in the vicinity of the physical interface with configuration ND and polynomial degrees $d = 3, 5$ than with configuration DN. From the previous explanation, better accuracy is obtained prescribing the Neumann boundary condition in the subproblem with the smoothest analytic solution, which in this case corresponds to subproblem P^{A}. Indeed, configuration DN prescribes the Neumann boundary condition in subproblem P^{A}, which has a smoother analytic solution than subproblem P^{B}. On the contrary, configuration ND provides worst accuracy than configuration DN since the Neumann boundary condition is prescribed in subproblem P^{B}. The results from this extreme case, where the analytic solution of subproblem P^{A} is much smoother than the analytic solution of subproblem P^{B}, further reinforces the assumed explanation and, therefore, approximate solution accuracy benefits from imposing the boundary conditions in the

subproblems according to the associated analytic solutions regularity. However, there is a trade-off between several parameters and conclusions should be drawn carefully for each situation.

Table 3.7: Relative errors and convergence orders obtained in the rose-shaped interface with the continuity interface conditions test case with a high thermal conductivity ratio and uniform Delaunay triangular meshes.

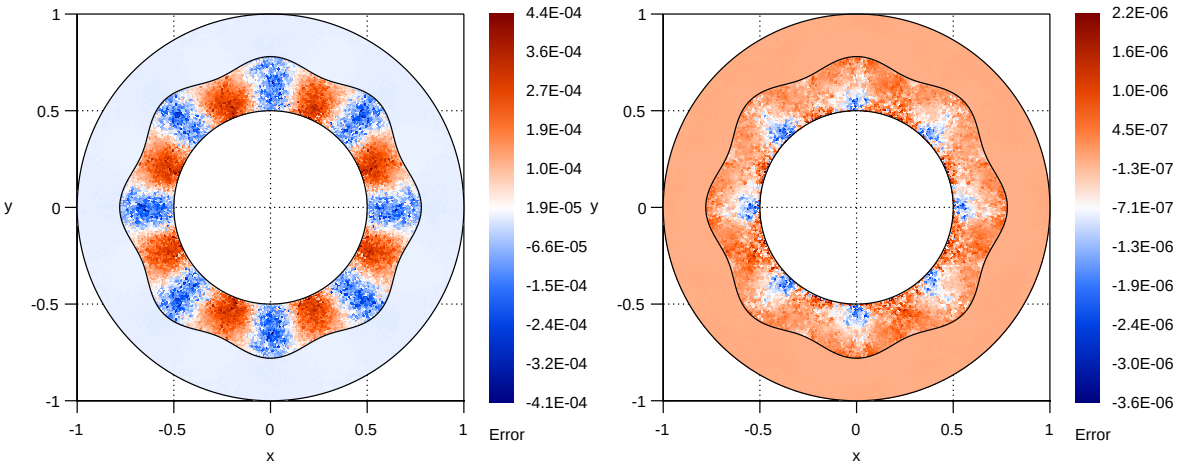
	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Configuration DN												
1 132	9.00E−04	–	7.29E−03	–	4.74E−05	–	5.92E−04	–	2.26E−05	–	3.02E−04	–
4 548	2.28E−04	1.97	1.98E−03	1.88	4.06E−06	3.53	5.02E−05	3.55	3.12E−07	6.16	9.34E−06	5.00
18 124	5.50E−05	2.06	4.44E−04	2.16	1.76E−07	4.54	3.60E−06	3.81	4.74E−09	6.06	1.38E−07	6.09
71 844	1.51E−05	1.88	1.29E−04	1.79	1.28E−08	3.81	2.53E−07	3.85	6.66E−11	6.19	2.82E−09	5.65
289 954	3.62E−06	2.05	3.42E−05	1.90	8.18E−10	3.94	1.92E−08	3.70	1.22E−11	–	1.41E−10	–
Configuration ND												
1 132	7.44E−04	–	6.96E−03	–	9.37E−05	–	1.47E−03	–	2.48E−05	–	3.06E−04	–
4 548	1.86E−04	2.00	1.87E−03	1.89	6.83E−06	3.77	6.42E−05	4.50	5.11E−07	5.58	9.40E−06	5.01
18 124	4.56E−05	2.03	4.23E−04	2.15	2.96E−07	4.54	4.10E−06	3.98	7.62E−09	6.08	4.14E−07	4.52
71 844	1.20E−05	1.94	1.28E−04	1.73	1.96E−08	3.94	2.77E−07	3.92	1.04E−10	6.24	2.81E−09	7.25
289 954	2.79E−06	2.09	3.41E−05	1.90	1.25E−09	3.95	1.92E−08	3.83	1.71E−12	–	7.60E−11	–

The same test case is addressed with successive finer uniform structured quadrilateral meshes and the simulations are performed with the same polynomial degrees, partitioning configurations, and method, as previously. The measured relative errors and convergence orders are reported in Table 3.8.

The optimal second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms, and the results confirm once again that the proposed method is capable of handling different types of polygonal meshes preserving the optimal convergence orders. The obtained behaviour concerning configurations DN and ND follows the one obtained with uniform Delaunay triangular meshes. That is, configuration DN provides more accurate approximate solutions with polynomial degrees $d = 3, 5$ than configuration ND, both in terms of the L^1- and L^∞-norms, although the differences are not significant.

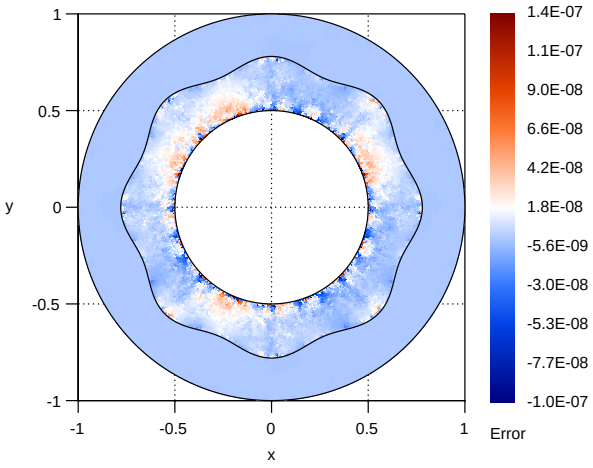
3.6.3 Circular interface with the imperfect interface conditions test case

The test case introduced in Section 3.6.1 is addressed prescribing the imperfect interface conditions instead of the continuity interface conditions and, therefore, the middle circumference is denoted as physical interface Γ^{K}. For the same analytic solutions, given in Equation (3.65), parameters a^{A}, a^{B}, b^{A}, and b^{B} are determined such that the Dirichlet boundary conditions and the imperfect interface conditions are simultaneously satisfied. Constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x})$ and $\kappa^{\mathrm{B}}(\boldsymbol{x})$ and constant interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x})$, for which the parameters are



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



(c) Polynomial degree $d = 5$.

Figure 3.25: Relative error distribution obtained in the rose-shaped interface with the continuity interface conditions test case with a high thermal conductivity ratio and configuration DN.

determined as

$$a^{\mathrm{A}} = -ch^{\mathrm{K}}\kappa^{\mathrm{B}}r_{\mathrm{M}}, \tag{3.87}$$

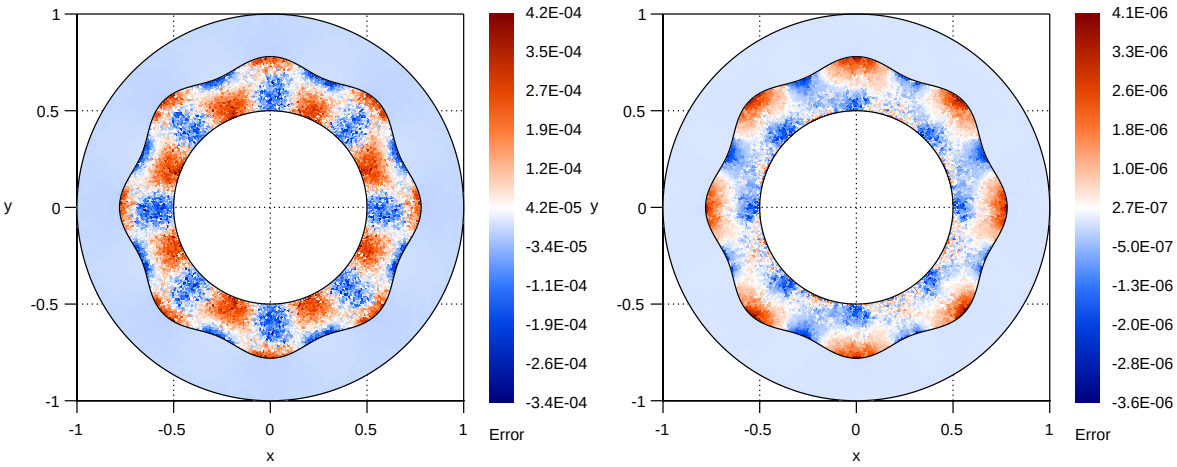
$$a^{\mathrm{B}} = -ch^{\mathrm{K}}\kappa^{\mathrm{A}}r_{\mathrm{M}}, \tag{3.88}$$

$$b^{\mathrm{A}} = c\left(h^{\mathrm{K}}r_{\mathrm{M}}\left(\kappa^{\mathrm{A}}\ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + \kappa^{\mathrm{B}}\ln(r_{\mathrm{M}})\right) - \kappa^{\mathrm{A}}\kappa^{\mathrm{B}}\right), \tag{3.89}$$

$$b^{\mathrm{B}} = ch^{\mathrm{K}}\kappa^{\mathrm{A}}r_{\mathrm{M}}\ln(r_{\mathrm{I}}), \tag{3.90}$$

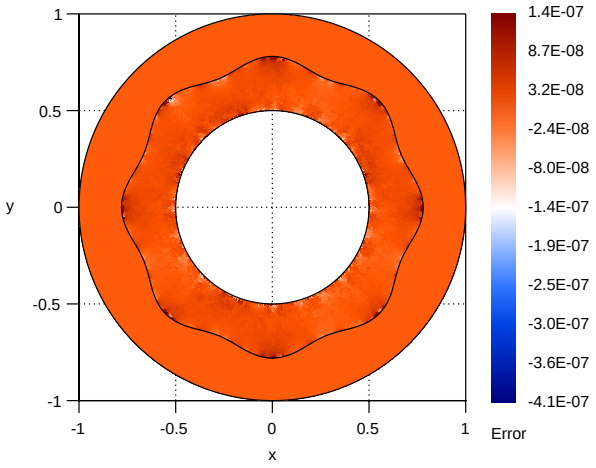
$$c = \frac{1}{h^{\mathrm{K}}r_{\mathrm{M}}\left(\kappa^{\mathrm{A}}\ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + \kappa^{\mathrm{B}}\ln\left(\frac{r_{\mathrm{M}}}{r_{\mathrm{E}}}\right)\right) - \kappa^{\mathrm{A}}\kappa^{\mathrm{B}}}. \tag{3.91}$$

The low thermal conductivity ratio case with constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 2$



(a) Polynomial degree $d = 1$.

(b) Polynomial degree $d = 3$.



(c) Polynomial degree $d = 5$.

Figure 3.26: Relative error distribution obtained in the rose-shaped interface with the continuity interface conditions test case with a high thermal conductivity ratio and configuration ND.

and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 1$ and the high thermal conductivity ratio case with constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 100$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 1$ are addressed. Both cases are also addressed with interfacial thermal conductance functions $h^{\mathrm{K}}(\boldsymbol{x}) = 100$, $h^{\mathrm{K}}(\boldsymbol{x}) = 1$, and $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$, where the higher thermal conductance recovers a nearly perfect thermal contact, whereas the lower thermal conductance recovers a nearly adiabatic thermal contact. The imperfect interface conditions imply a discontinuous temperature on the physical interface and the produced finite jump is inversely proportional to the associated interfacial thermal conductance function. The heat transfer between physical subdomains increases with increasing interfacial thermal conductance resulting in a stronger thermal coupling and, therefore, smaller temperature jump. For the low thermal conductivity ratio case, maximum temperature jumps on the physical interface for interfacial thermal conductance functions $h^{\mathrm{K}}(\boldsymbol{x}) = 100$, $h^{\mathrm{K}}(\boldsymbol{x}) = 1$, and $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$ are approximately 0.024, 0.71, and 1.00, respectively, while, for the high thermal conductivity ratio case, the maximum temperature jumps are approximately 0.032, 0.77, and 1.00, respectively. Notice that, as the interfacial thermal conductance function increases, the

Table 3.8: Relative errors and convergence orders obtained in the rose-shaped interface with the continuity interface conditions test case with a high thermal conductivity ratio and uniform structured quadrilateral meshes.

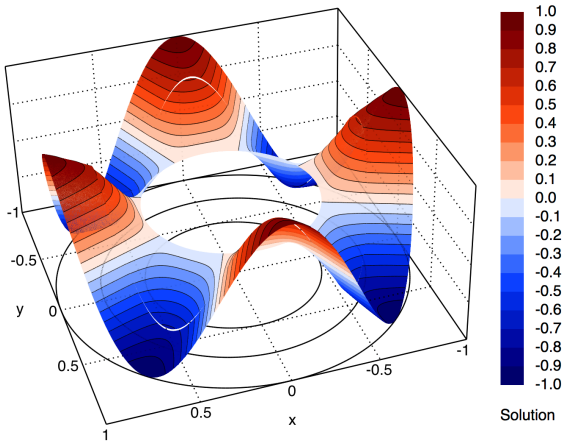
	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Configuration DN												
1 120	9.26E−04	–	4.43E−03	–	9.12E−05	–	7.72E−04	–	1.39E−04	–	7.62E−04	–
4 800	2.32E−04	1.90	1.18E−03	1.82	6.06E−06	3.73	5.60E−05	3.60	1.64E−06	6.10	2.01E−05	5.00
19 840	5.84E−05	1.94	3.00E−04	1.93	4.29E−07	3.73	3.81E−06	3.79	2.93E−08	5.68	4.30E−07	5.42
80 640	1.46E−05	1.97	7.54E−05	1.97	3.63E−08	3.52	2.68E−07	3.78	4.95E−10	5.82	7.21E−09	5.83
325 120	3.69E−06	1.98	1.88E−05	1.99	3.81E−09	3.23	2.61E−08	3.34	8.30E−12	5.86	1.14E−10	5.96
Configuration ND												
1 120	8.80E−04	–	4.76E−03	–	2.60E−04	–	2.29E−03	–	1.61E−04	–	9.35E−04	–
4 800	2.25E−04	1.87	1.32E−03	1.77	1.58E−05	3.86	1.89E−04	3.43	3.30E−06	5.34	3.58E−05	4.49
19 840	5.67E−05	1.94	3.35E−04	1.93	8.49E−07	4.12	1.09E−05	4.03	3.74E−08	6.31	4.31E−07	6.23
80 640	1.42E−05	1.97	8.46E−05	1.96	5.38E−08	3.94	5.94E−07	4.15	5.51E−10	6.02	7.20E−09	5.84
325 120	3.55E−06	1.99	2.13E−05	1.98	4.52E−09	3.55	3.06E−08	4.25	1.01E−11	5.73	1.74E−10	5.35

imperfect interface conditions approximate the continuity interface conditions. The resulting analytic solutions for the low and the high thermal conductivity ratio cases are shown in Figures 3.27 and the associated source terms are shown in Figures 3.28.

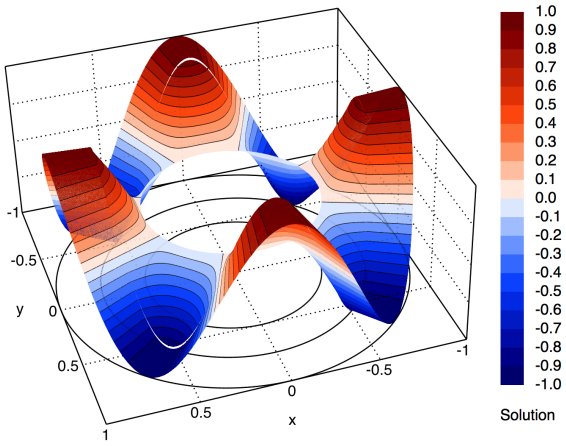
Simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1,3,5$, and Method 5 are carried out. The measured relative errors and convergence orders are reported in Table 3.9. Notice that only Neumann boundary conditions are prescribed on the interface in both subproblems.

For all the cases, the optimal second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1,3,5$, respectively, both in terms of the L^1- and L^∞-norms, similarly to the circular interface with the continuity interface conditions test case. Additionally, essentially non-oscillatory behaviour is observed and, for the same approximate solution, the measured relative errors in the L^∞-norm are, roughly speaking, at most a few tens times larger than those measured in the L^1-norm. The relatively low ratios indicate that no suspicious and critical errors are obtained in the vicinity of the interface and that the relative error distribution is relatively even in both physical subdomains.

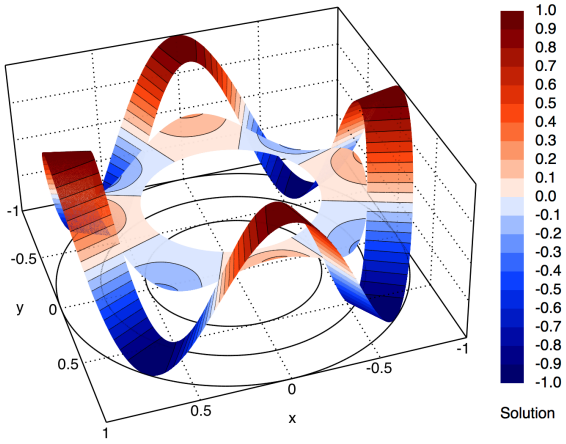
The results support the capability of the proposed method to handle imperfect interface conditions, low and high thermal conductivity ratios, and low and high interfacial thermal conductance functions, and provide very high-order of convergence. Moreover, the proposed method captures the discontinuous temperature on the interface, preserving the essentially non-oscillatory behaviour without difficulties and the optimal convergence orders are achieved.



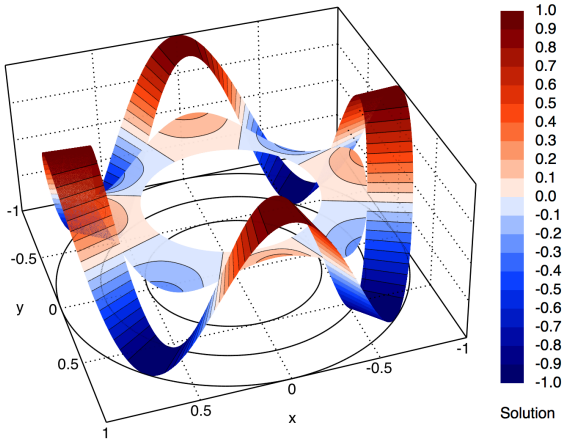
(a) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



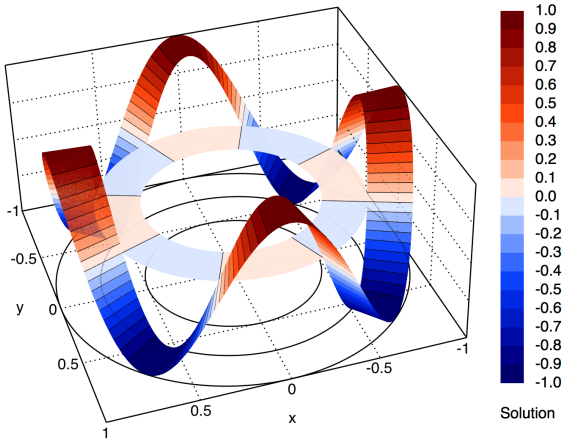
(b) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



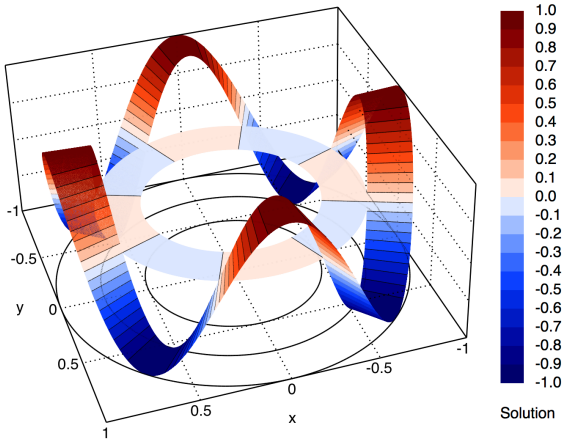
(c) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.



(d) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.

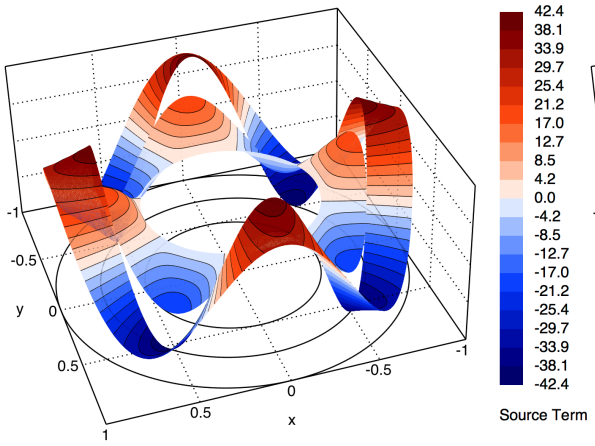


(e) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.

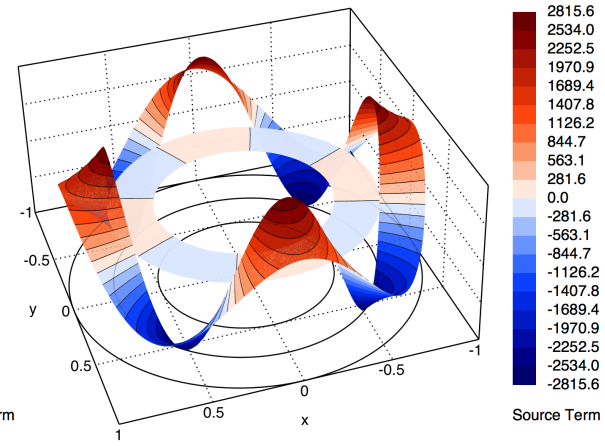


(f) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.

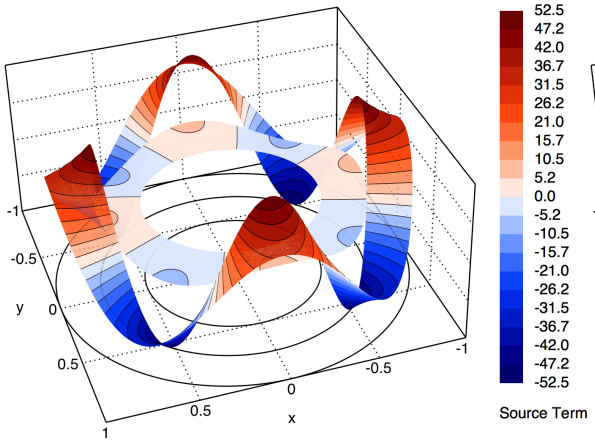
Figure 3.27: Analytic solutions for the circular interface with the imperfect interface conditions test case.



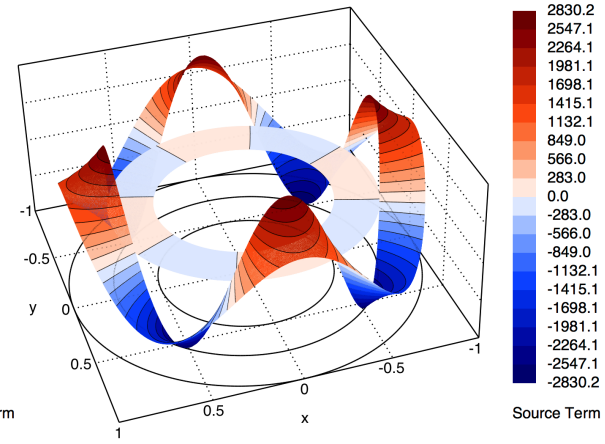
(a) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



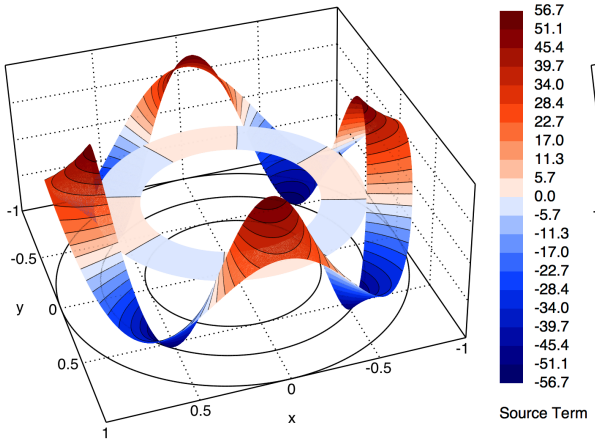
(b) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



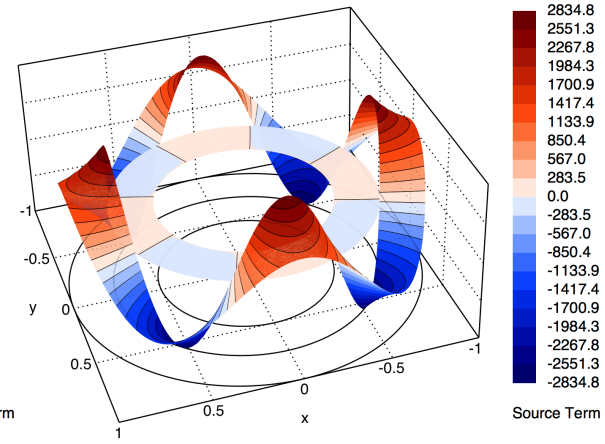
(c) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.



(d) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.



(e) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.



(f) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.

Figure 3.28: Source terms for the circular interface with the imperfect interface conditions test case.

Table 3.9: Relative errors and convergence orders obtained in the circular interface with the imperfect interface conditions test case with uniform Delaunay triangular meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=100$												
1 052	3.51E−03	–	8.33E−03	–	9.86E−05	–	5.45E−04	–	2.50E−05	–	1.54E−04	–
4 484	9.31E−04	1.83	2.74E−03	1.53	4.25E−06	4.34	2.14E−05	4.46	2.91E−07	6.14	1.79E−06	6.15
18 258	2.46E−04	1.89	8.10E−04	1.74	3.28E−07	3.65	1.58E−06	3.71	4.02E−09	6.10	3.01E−08	5.82
73 834	6.32E−05	1.95	2.58E−04	1.64	1.93E−08	4.05	1.36E−07	3.51	4.84E−11	6.33	4.25E−10	6.10
292 928	1.70E−05	1.90	6.73E−05	1.95	1.09E−09	4.18	8.95E−09	3.95	5.79E−12	–	8.65E−11	–
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=100$												
1 052	4.27E−03	–	1.29E−02	–	2.44E−04	–	1.45E−03	–	5.21E−05	–	3.18E−04	–
4 484	1.23E−03	1.72	4.09E−03	1.58	1.37E−05	3.97	6.49E−05	4.29	7.26E−07	5.89	3.85E−06	6.09
18 258	3.11E−04	1.96	9.80E−04	2.04	7.73E−07	4.09	3.88E−06	4.01	1.18E−08	5.87	8.10E−08	5.50
73 834	7.97E−05	1.95	2.93E−04	1.73	4.87E−08	3.96	4.31E−07	3.14	1.68E−10	6.08	1.45E−09	5.76
292 928	2.26E−05	1.83	9.97E−05	1.56	2.41E−09	4.36	1.69E−08	4.71	6.13E−12	–	6.08E−11	–
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=1$												
1 052	3.12E−03	–	8.87E−03	–	8.51E−05	–	3.24E−04	–	1.97E−05	–	6.93E−05	–
4 484	8.12E−04	1.86	3.07E−03	1.47	5.61E−06	3.75	2.82E−05	3.37	3.07E−07	5.74	1.54E−06	5.26
18 258	2.19E−04	1.86	9.04E−04	1.74	3.23E−07	4.06	1.82E−06	3.90	4.58E−09	5.99	4.38E−08	5.07
73 834	5.47E−05	1.99	2.71E−04	1.73	1.83E−08	4.11	1.88E−07	3.26	5.90E−11	6.23	4.92E−10	6.42
292 928	1.50E−05	1.88	8.87E−05	1.62	1.03E−09	4.18	8.57E−09	4.48	8.27E−12	–	1.27E−10	–
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=1$												
1 052	3.45E−03	–	1.03E−02	–	1.13E−04	–	4.08E−04	–	2.50E−05	–	9.82E−05	–
4 484	8.97E−04	1.86	3.42E−03	1.52	7.49E−06	3.74	4.06E−05	3.18	3.98E−07	5.71	2.19E−06	5.25
18 258	2.41E−04	1.87	9.83E−04	1.77	4.11E−07	4.14	2.47E−06	3.99	6.20E−09	5.93	5.79E−08	5.18
73 834	6.01E−05	1.99	2.94E−04	1.73	2.34E−08	4.10	2.51E−07	3.27	8.30E−11	6.17	7.09E−10	6.30
292 928	1.66E−05	1.87	1.00E−04	1.56	1.25E−09	4.26	9.29E−09	4.79	7.71E−12	–	6.35E−11	–
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=0.01$												
1 052	2.79E−03	–	1.03E−02	–	6.94E−05	–	4.14E−04	–	1.48E−05	–	9.88E−05	–
4 484	7.31E−04	1.85	3.39E−03	1.53	5.85E−06	3.41	4.08E−05	3.20	2.86E−07	5.44	2.21E−06	5.24
18 258	1.98E−04	1.86	9.82E−04	1.77	2.79E−07	4.33	2.47E−06	3.99	4.21E−09	6.01	5.83E−08	5.18
73 834	4.86E−05	2.01	2.94E−04	1.73	1.53E−08	4.16	2.53E−07	3.26	5.63E−11	6.18	7.08E−10	6.31
292 928	1.35E−05	1.85	1.01E−04	1.55	7.94E−10	4.30	9.30E−09	4.79	9.53E−12	–	1.33E−10	–
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=0.01$												
1 052	2.80E−03	–	1.03E−02	–	6.98E−05	–	4.12E−04	–	1.47E−05	–	9.89E−05	–
4 484	7.34E−04	1.85	3.42E−03	1.52	5.86E−06	3.42	4.09E−05	3.19	2.86E−07	5.43	2.21E−06	5.25
18 258	1.99E−04	1.86	9.85E−04	1.78	2.81E−07	4.33	2.49E−06	3.99	4.22E−09	6.01	5.82E−08	5.18
73 834	4.87E−05	2.01	2.95E−04	1.73	1.54E−08	4.16	2.53E−07	3.27	5.66E−11	6.17	7.12E−10	6.30
292 928	1.36E−05	1.85	1.01E−04	1.56	7.94E−10	4.30	9.30E−09	4.79	8.01E−12	–	6.17E−11	–

3.6.4 Rose-shaped interface with the imperfect interface conditions test case

As previously, the test case introduced in Section 3.6.2 is addressed prescribing the imperfect interface conditions instead of the continuity interface conditions and, therefore, the middle circumference is denoted as physical interface Γ^{K}. For the same analytic solutions, given in Equation (3.80), parameters a^{A}, a^{B}, b^{A}, and b^{B} are determined such that the Dirichlet boundary conditions and the imperfect interface conditions are simultaneously satisfied.

Contrarily to the circular interface test cases, the diffeomorphic transformation results in analytic solutions with a constant value on the physical interface, when applied to logarithm analytic solutions invariant by rotation. The same behaviour is observed replacing the continuity interface conditions with the imperfect interface condition, with the difference that a constant discontinuity is produced on the physical interface. Moreover, the diffeomorphic transformation preserves the conservation of the conductive heat flux on the physical interface. However, non-constant values are obtained contrarily to the analytic solutions. Therefore, the imperfect interface conditions impose that a constant temperature discontinuity and a constant interfacial thermal conductance function have to provide a non-constant conductive heat flux on the physical interface, which is not possible with Equation (3.5). Non-constant thermal conductivity functions are considered to remedy the issue in the physical subdomains such that the conductive heat flux on the physical interface is constant.

Thermal conductivity functions considered in each physical subdomain are given in polar coordinates as $\kappa^S(r,\theta) = s^S F_\kappa(\theta)$ with parameters $s^S \in \mathbb{R}^+$ and conduction factor function $F_\kappa(\theta)$ is given in polar coordinates as

$$F_\kappa(\theta) = \frac{-h^{\mathrm{K}}\left(\phi^{\mathrm{A}}(T(\theta),\theta) - \phi^{\mathrm{B}}(T(\theta),\theta)\right)}{\nabla\phi^{\mathrm{A}}(T(\theta),\theta)\cdot\boldsymbol{n}^{\mathrm{A}}(\theta)}, \tag{3.92}$$

and is shown in Figures 3.29. Notice that the thermal conductivity ratio $\kappa^{\mathrm{A}}(\theta)/\kappa^{\mathrm{B}}(\theta) = s^{\mathrm{A}}/s^{\mathrm{B}}$ is constant on the physical interface. Moreover, the variations of thermal conductivity functions $\kappa^{\mathrm{A}}(\theta)$ and $\kappa^{\mathrm{B}}(\theta)$ in the associated physical subdomains are, in fact, small since the variations in function $F_\kappa(\theta)$ are also small.

Constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x})$ and $\kappa^{\mathrm{B}}(\boldsymbol{x})$ and constant interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x})$ are addressed, for which the parameters are determined as

$$a^{\mathrm{A}} = -ch^{\mathrm{K}}s^{\mathrm{B}}r_{\mathrm{M}}, \tag{3.93}$$

$$a^{\mathrm{B}} = -ch^{\mathrm{K}}s^{\mathrm{A}}r_{\mathrm{M}}, \tag{3.94}$$

$$b^{\mathrm{A}} = c\left(h^{\mathrm{K}}r_{\mathrm{M}}\left(s^{\mathrm{A}}\ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + s^{\mathrm{B}}\ln(r_{\mathrm{M}})\right) - s^{\mathrm{A}}s^{\mathrm{B}}\right), \tag{3.95}$$

$$b^{\mathrm{B}} = ch^{\mathrm{K}}s^{\mathrm{A}}r_{\mathrm{M}}\ln(r_{\mathrm{I}}), \tag{3.96}$$

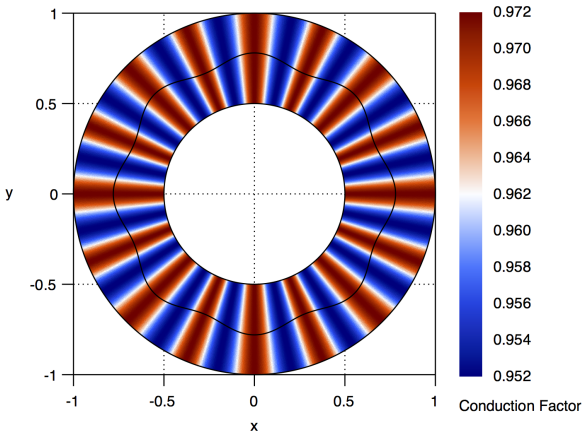


Figure 3.29: Function $F_\kappa(\theta)$ for the rose-shaped interface with the imperfect interface conditions test case.

$$c = \frac{1}{h^{\mathrm{K}} r_{\mathrm{M}} \left(s^{\mathrm{A}} \ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right) + s^{\mathrm{B}} \ln\left(\frac{r_{\mathrm{M}}}{r_{\mathrm{E}}}\right)\right) - s^{\mathrm{A}} s^{\mathrm{B}}}. \tag{3.97}$$

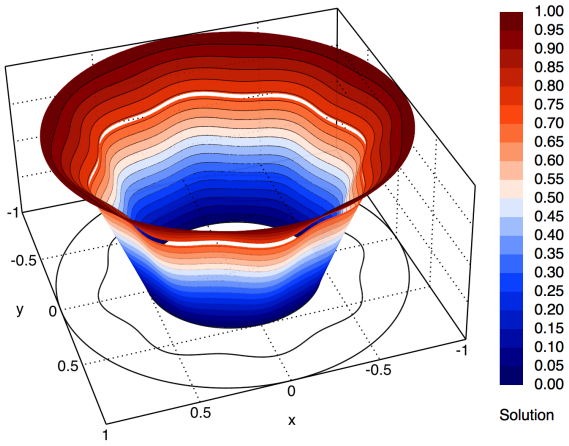
The low thermal conductivity ratio case, with $s^{\mathrm{A}} = 2$ and $s^{\mathrm{B}} = 1$, and the high thermal conductivity ratio case, with $s^{\mathrm{A}} = 100$ and $s^{\mathrm{B}} = 1$, are addressed for interfacial thermal conductance functions $h^{\mathrm{K}}(\boldsymbol{x}) = 100$, $h^{\mathrm{K}}(\boldsymbol{x}) = 1$, and $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$. An almost perfect thermal contact and a nearly adiabatic thermal contact are recovered for interfacial thermal conductance functions $h^{\mathrm{K}}(\boldsymbol{x}) = 100$ and $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$, respectively. Discontinuous temperatures on the physical interface are produced with the same maximum jumps as for the test case in Section 3.6.3. The resulting analytic solutions for the low and the high thermal conductivity ratio cases are shown in Figures 3.30 and the associated source terms are shown in Figures 3.31.

Simulations with successive finer uniform Delaunay triangular meshes, polynomial degrees $d = 1, 3, 5$, and Method 5 are carried out. The measured relative errors and convergence orders are reported in Table 3.10. Notice that only Neumann boundary conditions are prescribed on the interface in both subproblems.

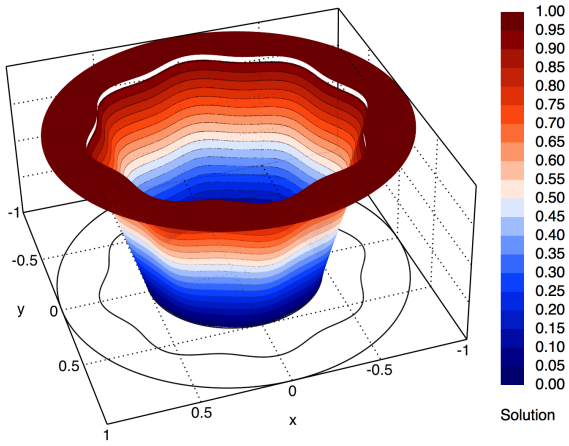
For all the cases, the optimal second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms, as for the circular interface with the continuity interface conditions test case. An essentially non-oscillatory behaviour is observed, and the ratios between the measured relative errors in the L^∞-norm and in the L^1-norm for the same approximate solution are, roughly speaking, at most a few tens. These ratios indicate that the relative error distribution is relatively even in both physical subdomains.

The results follow the previous test cases and support, once again, the capability of the proposed method to handle several situations of the conjugate heat transfer problem with imperfect interface conditions prescribed on complex curved physical interfaces and provide very high-order of convergence.

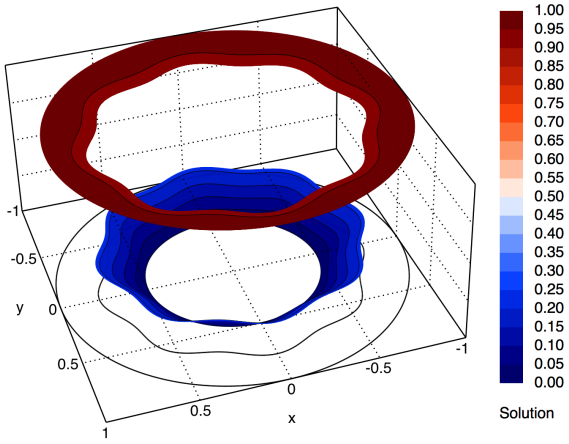
The same test case is addressed with successive finer uniform structured quadrilateral meshes



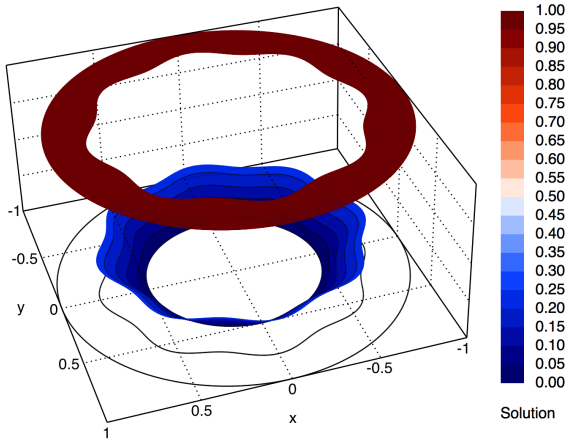
(a) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



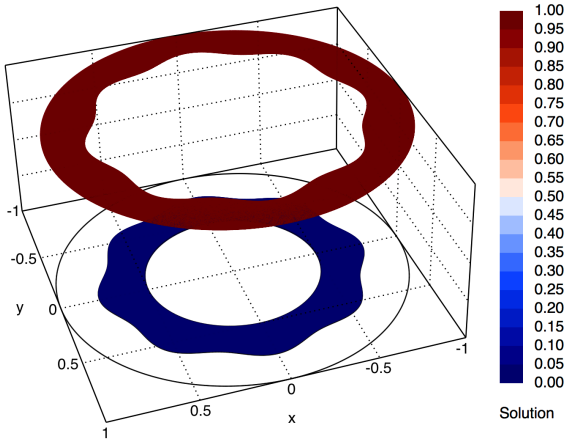
(b) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



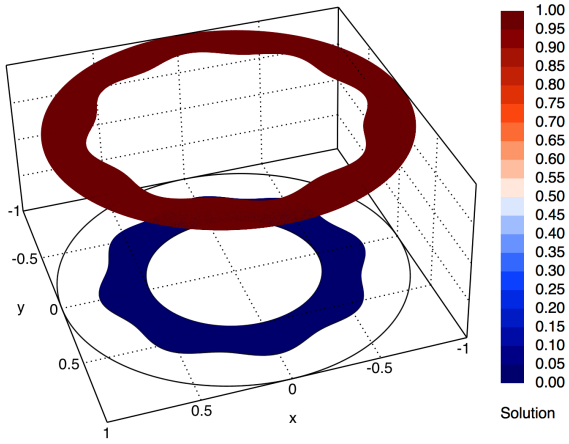
(c) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.



(d) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.

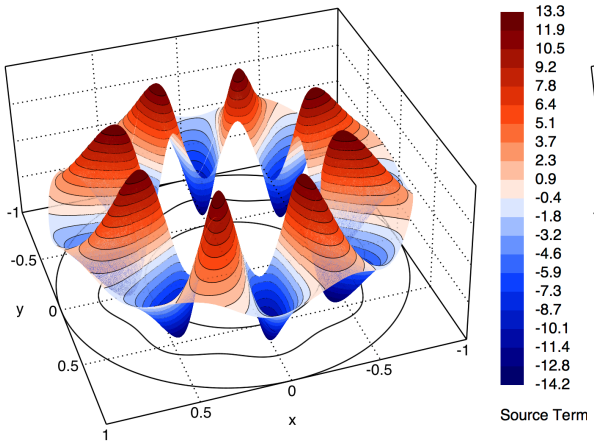


(e) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.

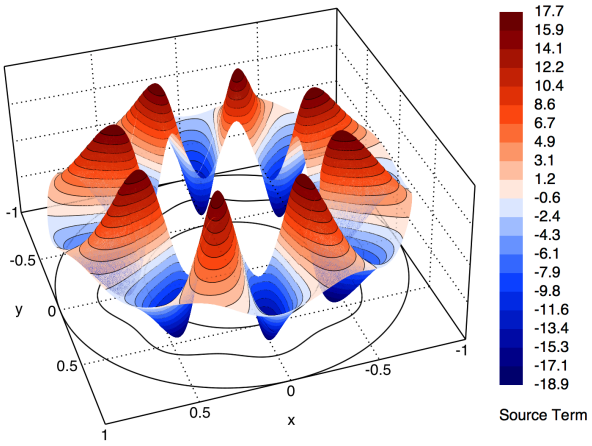


(f) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.

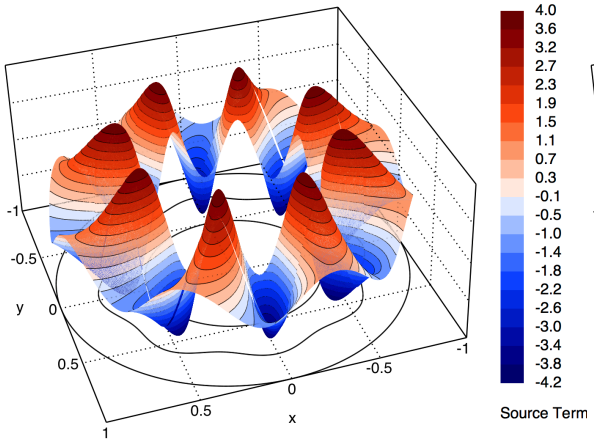
Figure 3.30: Analytic solutions for the rose-shaped interface with the imperfect interface conditions test case.



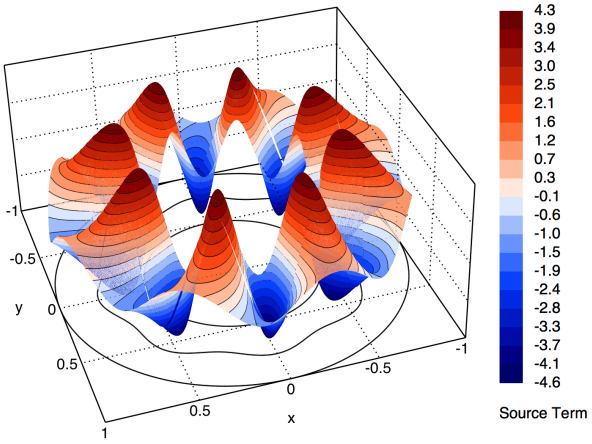
(a) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



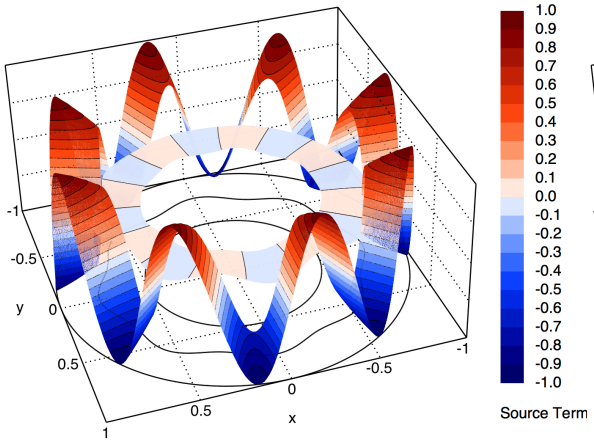
(b) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 100$.



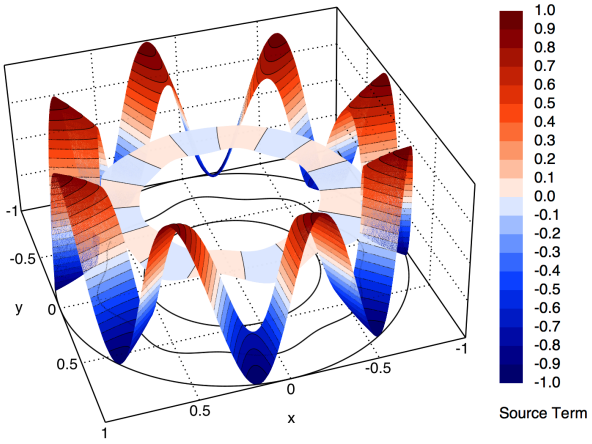
(c) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.



(d) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 1$.



(e) Low thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.



(f) High thermal conductivity ratio, interfacial thermal conductance function $h^{\mathrm{K}}(\boldsymbol{x}) = 0.01$.

Figure 3.31: Source terms for the rose-shaped interface with the imperfect interface conditions test case.

Table 3.10: Relative errors and convergence orders obtained in the rose-shaped interface with the imperfect interface conditions test case with uniform Delaunay triangular meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=100$												
1 132	1.37E−03	—	5.53E−03	—	9.15E−05	—	4.88E−04	—	5.13E−05	—	2.31E−04	—
4 548	3.34E−04	2.03	1.46E−03	1.91	6.23E−06	3.86	3.66E−05	3.73	7.68E−07	6.04	6.84E−06	5.06
18 124	8.39E−05	2.00	3.38E−04	2.12	2.70E−07	4.54	2.60E−06	3.82	8.29E−09	6.55	1.46E−07	5.56
71 844	2.23E−05	1.93	9.42E−05	1.85	1.70E−08	4.01	1.83E−07	3.85	8.87E−11	6.59	2.04E−09	6.20
289 954	5.40E−06	2.03	2.49E−05	1.91	1.17E−09	3.83	1.39E−08	3.70	1.04E−11	—	9.13E−11	—
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=100$												
1 132	7.29E−04	—	6.82E−03	—	8.74E−05	—	1.26E−03	—	2.32E−05	—	2.96E−04	—
4 548	1.82E−04	2.00	1.82E−03	1.90	6.49E−06	3.74	5.76E−05	4.43	4.68E−07	5.61	9.11E−06	5.01
18 124	4.43E−05	2.04	4.10E−04	2.15	2.77E−07	4.57	3.74E−06	3.95	7.05E−09	6.07	2.17E−07	5.40
71 844	1.17E−05	1.93	1.24E−04	1.73	1.85E−08	3.93	2.44E−07	3.96	9.76E−11	6.22	2.73E−09	6.36
289 954	2.74E−06	2.08	3.30E−05	1.90	1.20E−09	3.92	1.85E−08	3.69	1.09E−11	—	1.28E−10	—
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=1$												
1 132	4.18E−04	—	1.69E−03	—	2.79E−05	—	1.35E−04	—	1.54E−05	—	7.88E−05	—
4 548	1.04E−04	2.00	4.60E−04	1.87	2.05E−06	3.75	1.11E−05	3.59	2.83E−07	5.75	1.99E−06	5.29
18 124	2.52E−05	2.05	1.04E−04	2.15	8.27E−08	4.65	7.81E−07	3.84	2.67E−09	6.74	3.01E−08	6.07
71 844	6.73E−06	1.92	3.12E−05	1.75	5.57E−09	3.92	5.48E−08	3.86	2.81E−11	6.61	6.10E−10	5.66
289 954	1.65E−06	2.02	7.78E−06	1.99	3.96E−10	3.79	4.16E−09	3.69	7.92E−12	—	9.97E−11	—
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=1$												
1 132	2.72E−04	—	1.81E−03	—	1.41E−05	—	1.48E−04	—	6.37E−06	—	7.85E−05	—
4 548	6.83E−05	1.99	4.94E−04	1.87	1.57E−06	3.15	1.20E−05	3.61	1.45E−07	5.44	2.15E−06	5.17
18 124	1.54E−05	2.15	1.11E−04	2.15	4.63E−08	5.10	8.43E−07	3.85	1.46E−09	6.65	3.25E−08	6.06
71 844	4.19E−06	1.89	3.29E−05	1.77	4.16E−09	3.50	5.91E−08	3.86	1.90E−11	6.31	6.60E−10	5.66
289 954	1.03E−06	2.02	8.30E−06	1.97	3.27E−10	3.65	4.50E−09	3.69	9.75E−12	—	1.31E−10	—
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=0.01$												
1 132	2.51E−05	—	1.48E−04	—	4.73E−07	—	2.67E−06	—	2.24E−07	—	1.27E−06	—
4 548	5.11E−06	2.29	2.06E−05	2.84	3.24E−08	3.85	1.57E−07	4.07	4.61E−09	5.59	2.77E−08	5.50
18 124	1.09E−06	2.24	4.44E−06	2.22	1.27E−09	4.68	1.10E−08	3.84	3.95E−11	6.88	4.52E−10	5.95
71 844	2.62E−07	2.06	1.02E−06	2.13	8.57E−11	3.92	7.70E−10	3.86	2.84E−12	3.82	4.78E−11	3.26
289 954	7.12E−08	1.87	2.58E−07	1.98	6.29E−12	3.75	5.85E−11	3.69	7.59E−12	—	1.30E−10	—
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=0.01$												
1 132	4.19E−06	—	2.39E−05	—	1.91E−07	—	2.46E−06	—	9.31E−08	—	1.27E−06	—
4 548	1.04E−06	2.01	6.60E−06	1.85	2.05E−08	3.21	1.57E−07	3.95	2.57E−09	5.16	2.78E−08	5.50
18 124	2.27E−07	2.20	1.49E−06	2.16	7.35E−10	4.82	1.10E−08	3.85	2.25E−11	6.85	4.53E−10	5.95
71 844	7.83E−08	1.55	5.30E−07	1.50	5.46E−11	3.78	7.70E−10	3.86	2.39E−12	3.26	3.67E−11	3.65
289 954	2.35E−08	1.73	1.73E−07	1.61	5.13E−12	3.39	5.85E−11	3.69	6.37E−12	—	1.06E−10	—

considering the same polynomial degrees and Method 5, as previously. The measured relative errors and convergence orders are reported in Table 3.11.

The behaviour of the proposed method with uniform structured quadrilateral meshes is similar to the previous situation with uniform Delaunay triangular meshes in terms of convergence order. Therefore, the optimal second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms, and the results support the capability and versatility of the proposed method to provide very high-order of convergence with general polygonal meshes.

3.7 Conclusions

A very high-order accurate finite volume scheme is proposed to solve two-dimensional steady-state conjugate heat transfer problems in arbitrary curved domains with the continuity and imperfect interface conditions prescribed on arbitrary curved interfaces. The problem also consists of discontinuous velocity, thermal conductivity, source term, and temperature.

A partitioning method is applied to the conjugate heat transfer problem resulting in a partitioned problem that consists of Dirichlet and Neumann boundary value subproblems in the physical subdomains. The Dirichlet-Neumann and Neumann-Neumann methods are applied to the continuity and imperfect interface conditions, respectively. The thermal coupling between the subproblems consists in providing the appropriate functions associated with the produced boundary conditions such that the solution of the partitioned problem satisfies the interface conditions and, therefore, corresponds to the solution of the conjugate heat transfer problem. The polynomial reconstruction method is used to provide accurate local approximations of the unknown temperature functions in general polygonal meshes. Specifically constrained polynomial reconstructions associated with the boundary edges are computed to fulfil the prescribed boundary conditions in the subproblems, and the ROD method is used to provide the appropriate linear constraints for the associated minimization procedure in the least-squares sense. A finite volume method is used to discretize the subproblems, and the very high-order accurate scheme consists in determining the numerical convective and conductive heat fluxes on the straight edges evaluating the computed polynomial reconstructions for the subproblems. The system of linear equations for the partitioned problem is given in terms of affine residual operators, and a static condensation method is used to provide the approximate solution in terms of only internal variables.

A comprehensive numerical benchmark is provided and comprises test cases for conjugate heat transfer problems with the continuity and imperfect interface conditions prescribed on circular and rose-shaped interfaces. Several thermal conductivity functions for the physical subdomains and interfacial thermal conductance functions for the physical interface are considered to assess the capability of the proposed method in terms of accuracy, convergence order, stability, and robustness.

Table 3.11: Relative errors and convergence orders obtained in the rose-shaped interface with the imperfect interface conditions test case with uniform structured quadrilateral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio, $h_{\mathrm{K}}(\boldsymbol{x})=100$												
1 120	1.36E−03	–	3.75E−03	–	2.28E−04	–	1.29E−03	–	1.81E−04	–	1.09E−03	–
4 800	3.35E−04	1.93	9.32E−04	1.91	1.52E−05	3.72	1.30E−04	3.15	3.06E−06	5.60	1.69E−05	5.73
19 840	8.31E−05	1.96	2.38E−04	1.92	8.84E−07	4.01	7.38E−06	4.05	3.54E−08	6.29	3.12E−07	5.62
80 640	2.07E−05	1.98	6.02E−05	1.96	5.68E−08	3.92	3.57E−07	4.32	7.42E−10	5.51	6.66E−09	5.48
325 120	5.17E−06	1.99	1.52E−05	1.98	5.87E−09	3.26	2.13E−08	4.04	1.10E−11	6.05	9.29E−11	6.13
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=100$												
1 120	2.78E−04	–	1.40E−03	–	2.40E−04	–	2.03E−03	–	1.58E−04	–	7.87E−04	–
4 800	6.90E−05	1.91	3.41E−04	1.94	1.47E−05	3.84	1.71E−04	3.40	3.11E−06	5.40	2.88E−05	4.54
19 840	1.72E−05	1.96	8.43E−05	1.97	7.99E−07	4.10	9.60E−06	4.06	3.67E−08	6.26	4.17E−07	5.97
80 640	4.29E−06	1.98	2.09E−05	1.99	5.05E−08	3.94	5.20E−07	4.16	5.16E−10	6.08	6.98E−09	5.84
325 120	1.06E−06	2.00	5.21E−06	2.00	4.32E−09	3.53	2.71E−08	4.24	9.68E−12	5.70	1.31E−10	5.70
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=1$												
1 120	4.05E−04	–	1.38E−03	–	6.15E−05	–	2.36E−04	–	7.14E−05	–	3.22E−04	–
4 800	1.00E−04	1.92	3.36E−04	1.94	3.42E−06	3.97	1.31E−05	3.97	1.08E−06	5.76	4.88E−06	5.76
19 840	2.48E−05	1.97	8.21E−05	1.98	2.26E−07	3.83	1.04E−06	3.58	1.72E−08	5.83	9.39E−08	5.57
80 640	6.17E−06	1.98	2.04E−05	1.99	1.76E−08	3.64	1.15E−07	3.13	2.45E−10	6.06	1.56E−09	5.84
325 120	1.54E−06	1.99	5.10E−06	1.99	1.79E−09	3.28	9.95E−09	3.52	4.13E−12	5.86	3.40E−11	5.49
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=1$												
1 120	2.16E−04	–	1.20E−03	–	4.33E−05	–	2.47E−04	–	6.30E−05	–	3.37E−04	–
4 800	5.82E−05	1.80	3.52E−04	1.69	2.17E−06	4.11	1.58E−05	3.78	6.17E−07	6.36	4.73E−06	5.86
19 840	2.09E−05	1.44	1.72E−04	1.01	1.40E−07	3.87	1.27E−06	3.55	1.42E−08	5.31	1.02E−07	5.41
80 640	9.34E−06	1.15	8.19E−05	1.06	1.31E−08	3.38	1.32E−07	3.23	1.46E−10	6.53	1.69E−09	5.84
325 120	4.12E−06	1.17	2.50E−05	1.70	1.36E−09	3.25	1.11E−08	3.56	3.55E−12	5.33	3.76E−11	5.46
Low thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=0.01$												
1 120	9.43E−06	–	2.23E−05	–	1.26E−06	–	4.92E−06	–	1.22E−06	–	5.05E−06	–
4 800	2.31E−06	1.93	5.27E−06	1.98	7.43E−08	3.89	3.03E−07	3.83	1.72E−08	5.85	7.48E−08	5.79
19 840	5.73E−07	1.97	1.48E−06	1.79	4.09E−09	4.09	2.32E−08	3.62	3.09E−10	5.66	1.34E−09	5.67
80 640	1.40E−07	2.01	3.87E−07	1.92	3.04E−10	3.71	2.22E−09	3.34	3.88E−12	6.24	2.24E−11	5.84
325 120	3.46E−08	2.00	1.49E−07	1.37	2.72E−11	3.46	1.67E−10	3.71	7.78E−13	–	1.12E−11	–
High thermal conductivity ratio, $h^{\mathrm{K}}(\boldsymbol{x})=0.01$												
1 120	3.89E−06	–	2.09E−05	–	9.04E−07	–	4.93E−06	–	9.97E−07	–	5.06E−06	–
4 800	9.67E−07	1.91	5.15E−06	1.92	4.51E−08	4.12	3.04E−07	3.83	7.96E−09	6.64	7.47E−08	5.79
19 840	2.52E−07	1.89	1.27E−06	1.97	2.29E−09	4.20	2.32E−08	3.63	2.13E−10	5.10	1.34E−09	5.66
80 640	6.36E−08	1.96	3.02E−07	2.05	1.99E−10	3.48	2.22E−09	3.35	2.05E−12	6.62	2.21E−11	5.86
325 120	2.60E−08	1.28	1.13E−07	1.40	1.95E−11	3.34	1.64E−10	3.74	1.06E−12	–	8.66E−12	–

Analytic solutions are provided, satisfying the prescribed boundary, and interface conditions and the associated source terms are determined following the method of manufactured solutions. The proposed method, in addition to some alternative methods, to prove the need for specific treatment of curved domains, and different partitioning configurations, in the case of the continuity interface conditions, are addressed. The method of verification consists in determining the relative errors and the convergence orders in different norms obtained from the approximate solutions in successive finer meshes. Firstly, the results obtained from the alternative methods prove that, indeed, a specific treatment of the curved domains is necessary to preserve the optimal convergence orders of the method. On the other side, the proposed method achieves very high-order of convergence without non-physical oscillations in all the cases considered, even for discontinuous temperature on the physical interface. The optimal convergence orders are achieved when the constrained polynomial reconstructions associated with the boundary edges discretizing the physical interface have degree $d+1$ while the remaining have degree d. Concerning the partitioning configurations in the case of the continuity interface conditions, the results indicate that prescribing the Neumann condition in the subproblem with the smoothest solution provides more accurate approximate solutions. However, both partitioning configurations achieve the optimal convergence orders, and the differences in terms of accuracy only become significant when the thermal conductivity ratio increases. Therefore, based on the results, there is no particular reason for much concern about the partitioning configuration used.

The proposed method proved to be notorious in terms of its capability to solve conjugate heat transfer problems with intricate and complex phenomena on curved physical interfaces and provide very high-order of convergence. The proposed method is also very versatile and promising for further application in three-dimensional unsteady conjugate heat transfer problems and multiphysics problems in fluid mechanics, solid mechanics, and electromagnetics, aiming towards complex engineering applications. Moreover, non-smooth physical interfaces comprising geometrical singularities, such as sharp-edges, cusps, and tips, can be encountered in engineering applications and can result in numerical challenges that need to be addressed to properly preserve the accuracy, convergence order, stability, and robustness of the proposed method. Verification and validation in these scenarios will be required.

References

[1] C.A. Felippa, K.C. Park, C. Farhat, Staggered transient analysis procedures for coupled mechanical systems: formulation, Comput. Meth. Appl. Mech. Engrg. 24 (1980) 61–111.

[2] C.A. Felippa, K.C. Park, C. Farhat, Partitioned analysis of coupled mechanical systems, Comput. Meth. Appl. Mech. Engrg. 190(24–25) (2001) 3247–3270.

[3] P. Birken, K.J. Quint, S. Hartmann, A. Meister, A time-adaptive fluid-structure interaction method for thermal coupling, Comput. Visual Sci. 13 (2010) 331–340.

[4] B.A. Miller, J.J. McNamara, Loosely coupled time-marching of fluid-thermal-structural interactions with time-accurate CFD, 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2015.

[5] P.L. Kapitza, The study of heat transfer in helium II, J. Phys. (USSR) 4 (1941) 181–210.

[6] L. Fradette, P.A. Tanguy, P. Hurez, D. Blouin, On the determination of heat transfer coefficient between PVC and steel in vacuum extrusion calibrators, Int. J. Numer. Meth. Heat Fluid Flow 6(1) (1996) 3–12.

[7] A. Javili, S. Kaessmair, P. Steinmann, General imperfect interfaces, Comput. Methods Appl. Mech. Engrg. 275 (2014) 76–97.

[8] 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.

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

[10] 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.

[11] C. Farhat, K.G. Van der Zee, P. Geuzaine, Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, Comput. Methods Appl. Mech. Engrg. 195 (17-18) (2006) 1973–2001.

[12] V. Kazemi-Kamyab, A.H. van Zuijlen, H. Bijl, A high order time-accurate loosely-coupled solution algorithm for unsteady conjugate heat transfer problems, Comput. Meth. Appl. Mech. Engrg. 264 (2013) 205–217.

[13] V. Kazemi-Kamyab, A.H. van Zuijlen, H. Bijl, Accuracy and stability analysis of a second-order time-accurate loosely coupled partitioned algorithm for transient conjugate heat transfer problems, Int. J. Numer. Meth. Fluids 74 (2014) 113–133.

[14] V. Kazemi-Kamyab, A.H. Van Zuijlen, H. Bijl, Analysis and application of high order implicit Runge-Kutta schemes for unsteady conjugate heat transfer: a strongly-coupled approach, J. Comput. Phys. 272 (2014), 471–486.

[15] R. Lan, P. Sun, M. Mu, Mixed finite element analysis for an elliptic/mixed elliptic interface problem with jump coefficients, Procedia Comput. Sci. 108 (2017) 1913–1922.

[16] X. Pan, C. Lee, J.-I. Choi, Efficient monolithic projection method for time-dependent conjugate heat transfer problems, J. Comput. Phys. 369 (2018) 191–208.

[17] M.B. Giles, Stability analysis of numerical interface conditions in fluidâŞstructure thermal analysis, Int. J. Numer. Meth. Fluids 25(4) (1997) 421–436.

[18] F.-X. Roux, Domain decomposition methodology with Robin interface matching conditions for solving strongly coupled problems, in International Conference on Computational Science, Springer, Berlin, Heidelberg, 2008.

[19] W.D. Henshaw, K.K. Chand, A composite grid solver for conjugate heat transfer in fluidâŞstructure systems, J. Comput. Phys. 228 (2009) 3708–3741.

[20] M.-P. Errera, S. Chemin, Optimal solutions of numerical interface conditions in fluid-structure thermal analysis, J. Comput. Phys. 245 (2013) 431–455.

[21] O. Joshi, P. Leyland Stability analysis of a partitioned fluid–structure thermal coupling algorithm, J. Thermophis. Heat Tr. 28(1) (2014) 59–67.

[22] M.-P. Errera, F. Duchaine, Comparative study of coupling coefficients in Dirichlet-Robin procedure for fluid-structure aerothermal simulations, J. Comput. Phys. 312 (2016) 218–234.

[23] Meng, J.W. Banks, W.D. Henshaw, D.W. Schwendeman, A stable and accurate partitioned algorithm for conjugate heat transfer, J. Comput. Phys. 344 (2017) 51–85.

[24] 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.

[25] C. Dopazo, On conditioned averages for intermittent turbulent flows, J. Fluid Mech. 81 (3) (1977) 433–438.

[26] F. Habla, L. Dietsche, O. Hinrichsen, Modeling and simulation of conditionally volume averaged viscoelastic two-phase flows, AIChE Journal, 59 (10) (2013) 3914–3927.

[27] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.-O. Persson, B. Leer, M. Visbal, High-order CFD methods: current status and perspective, Int. J. Numer. Meth. Fluids 72 (2013) 811–845.

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

[29] Z. Cai, B. Thornber, An internal penalty discontinuous Galerkin method for simulating conjugate heat transfer in a closed cavity, Int. J. Numer. Meth. Fluids 87(3) (2018) 134-159.

[30] A. Cangiani, E. Georgoulis, Y. Sabawi, Adaptive discontinuous Galerkin methods for elliptic interface problems, Math. Comput. 87(314) (2018) 2675–2707.

[31] L.N.T. Huynh, N.C. Nguyen, J. Peraire, B.C. Khoo, A high-order hybridizable discontinuous Galerkin method for elliptic interface problems, Int. J. Numer. Meth. Engrg. 93(2) (2013) 183–200.

[32] W. Qiu, M. Solano, P. Vega, A high order HDG method for curved-interface problems via approximations from straight triangulations, J. Sci. Comput. 69(3) (2016) 1384–1407.

[33] M. Paipuri, C. Tiago, S. Fernández-Méndez, Coupling of continuous and hybridizable discontinuous Galerkin methods: application to conjugate heat transfer problem, J. Sci. Comput. 78 (2019) 321–350.

[34] B. Wang, B.C. Khoo, Hybridizable discontinuous Galerkin method (HDG) for Stokes interface flow, J. Comput. Phys. 247 (2013) 262–278.

[35] H. Sun , D.L. Darmofal, An adaptive simplex cut-cell method for high-order discontinuous Galerkin discretizations of elliptic interface problems and conjugate heat transfer problems, J. Comput. Phys. 278 (2014) 445–468.

[36] S.M. Ojeda, H. Sun, S.R. Allmaras. D.L. Darmofal, An adaptive simplex cut-cell method for high-order discontinuous Galerkin discretizations of conjugate heat transfer problems, Int. J. Numer. Meth. Engrg. 110 (2017) 350–378.

[37] L. Mu, J. Wang, G. Wei, X. Ye, S. Zhao, Weak Galerkin methods for second order elliptic interface problems, J. Comput. Phys. 250 (2013) 106–125.

[38] L. Mu, J. Wang, X. Ye, S. Zhao, A new weak Galerkin finite element method for elliptic interface problems, J. Comput. Phys. 325 (2016) 157–173.

[39] J. Guzmán, M.A. Sánchez, M. Sarkis, Higher-order finite element methods for elliptic problems with interfaces, ESAIM: Math. Model. Numer. Anal. 50(5) (2016) 1561–1583.

[40] P. Huang, H. Wu, Y. Xiao, An unfitted interface penalty finite element method for elliptic interface problems, Comput. Meth. Appl. Mech. Engrg. 323 (2017) 439–460.

[41] 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.

[42] H. Ji, J. Chen, Z. Li, A high-order source removal finite element method for a class of elliptic interface problems, Appl. Numer. Math. 130 (2018) 112–130.

[43] H. Wu, Y. Xiao, An unfitted hp-interface penalty finite element method for elliptic interface problems, J. Comput. Math. 37(3) (2019) 316–339.

[44] J. Cheung, M. Perego, P. Bochev, M. Gunzburger, Optimally accurate high-order finite element methods for polytopial approximations of domains with smooth boundaries, Math. Comp. 88 (2019) 2187–2219.

[45] J. Cheung, M. Perego, P. Bochev, M. Gunzburger, An optimally convergent coupling approach for interface problems approximated with higher-order finite elements, arXiv e-prints, page arXiv:1710.01667, Oct 2017.

[46] Y.C. Zhou, S. Zhao, M. Feig, G.W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213(1) (2006) 1–30.

[47] X. Zhong, A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity, J. Comput. Phys. 225(1) (2007) 1066–1099.

[48] I.T. Angelova, L.G. Vulkov, High-order finite difference schemes for elliptic problems with intersecting interfaces, Appl. Math. Comput. 187(2) (2007) 824–843.

[49] A.N. Marques, J.-C. Nave, R.R. Rosales, High order solution of Poisson problems with piecewise constant coefficients and interface jumps, J. comput. Phys. 335 (2017) 497–515.

[50] J.-K. Liu, Z.-S. Zheng, Efficient high-order immersed interface methods for heat equations with interfaces, Appl. Math. Mech. 35(9) (2014) 1189–1202.

[51] M. Elhaddad, N. Zander, T. Bog, L. Kudela, S. Kollmannsberger, J. Kirschke, T. Baum, M. Ruess, E. Rank, Multi-level hp-finite cell method for embedded interface problems with application in biomechanics, Int. J. Numer. Meth. Biomed. Engrg. 34(4) (2018) e2951.

[52] 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.

[53] 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.

[54] D.Y. Kwak, S. Lee, Y. Hyon, A new finite element for interface problems having Robin type jump, Int. J. Numer. Anal. Model. 14(4–5) (2017) 532–549.

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

[56] 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.

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

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

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

[60] C. Ollivier-Gooch, A. Nejat, C. Michalak, On obtaining high-order finite-volume solutions to the Euler equations on unstructured meshes, AIAA Paper 2007-4464 (2007).

[61] C. Ollivier-Gooch, A. Nejat, C. Michalak, Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations, AIAA Journal 47 (9) (2009) 2105–2120.

[62] C. Michalak, C. Ollivier-Gooch, Unstructured high-order accurate finite volume solutions of the Navier-Stokes equations, AIAA Paper 2009-954 (2009).

[63] A. Boularas, S. Clain, F. Baudoin, A sixth-order finite volume method for diffusion problem with curved boundaries, Appl. Math. Model. 42 (2017) 401–422.

[64] 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.

[65] R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, R. Loubère, Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries, Int. J. Numer. Meth. Engrg. 117 (2) (2019) 188–220.

[66] S. Clain, G.J. Machado, J.M. Nóbrega, R.M.S. Pereira, A sixth-order finite volume method for the convection-diffusion problem with discontinuous coefficients, Comput. Meth. Appl. Mech. Engrg. 267 (2013) 43–64.

[67] T.J. Barth, P.O. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper 90-0013 (1990).

[68] T.J. Barth, Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations, in Unstructured Grid Methods for Advection-Dominated Flows, AGARD, AGARD-R-787 (1992).

[69] T.J. Barth, Recent developments in high order k-exact reconstruction on unstructured meshes, AIAA Paper 93-0668 (1993).

[70] S. Clain, G.J. Machado, A very high-order finite volume method for the time-dependent convection-diffusion problem with Butcher tableau extension, Comput. Math. Appl. 68 (2014) 1292–1311.

[71] R. Costa, S. Clain, G.J. Machado, R. Loubère, A very high-order accurate staggered finite volume scheme for the stationary incompressible Navier-Stokes and Euler equations on unstructured meshes, J. Sci. Comput. 71 (3) (2017) 1375–1411.

[72] R. Costa, S. Clain, G.J. Machado, A sixth-order finite volume scheme for the steady-state incompressible Stokes equations on staggered unstructured meshes, J. Comput. Phys. 349 (2017) 501–527.

[73] D.E. Wells, E.J. Krakiwsky, The method of least-squares, lecture notes 18, Department of Surveying Engineering, University of New Brunswick, Fredericton (1971).

[74] D.P. Bertsekas, Constrained optimization and Lagrange multiplier methods, 1st Edition, Academic Press (1982).

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

[76] P.J. Roache, Code verification by the method of manufactured solutions, J. Fluids Engrg. 124 (1) (2002) 4–10.

[77] C.J. Roy, Grid convergence error analysis for mixed-order numerical schemes, AIAA Journal 41 (4) (2003) 595–604.

[78] C.J. Roy, C.C. Nelson, T.M. Smith, C.C. Ober, Verification of Euler/Navier-Stokes codes using the method of manufactured solutions, Int. J. Numer. Meth. Fluids 44 (2004) 599–620.

[79] A. Veeraragavan, J.Beri, R.J.Gollan, Use of the method of manufactured solutions for the verification of conjugate heat transfer solvers, J. Comput. Phys. 307 (2016) 308–320.

CHAPTER 4

Thermoplastic sheet extrusion cooling stage simulation with a very high-order accurate finite volume scheme

Abstract: Manufacturing technologies in the polymer processing industry demand rigorous control of the involved physical variables to ensure that the produced parts meet the required specifications. The numerical modelling is nowadays an indispensable tool for the engineering design of such manufacturing technologies and the understanding of the complex physical phenomena acting in the process. In the present work, a numerical approach is proposed to investigate the complex heat exchanges in the thermoplastic sheet extrusion cooling stage, which requires solving a conjugate heat transfer problem between a thermoplastic sheet and a metallic roll. The discretization method is based on a finite volume method capable of providing a very high-order of convergence to achieve the required accuracy of the approximate solution with the lowest possible computational cost. A code verification benchmark based on manufactured solutions proves that the discretization method provides the optimal convergence orders and an improved trade-off between accuracy and computational cost is achieved with higher-orders of convergence. The capabilities of the proposed numerical approach are assessed by running several simulations for the sheet extrusion cooling stage and comparing the performance according to the convergence order of the discretization method and the aspect ratio of the mesh. The results prove that higher-orders of convergence are computationally more cost-effective than the classical second-order of convergence, and higher aspect ratios allow to improve the calculation efficiency further.

Keywords: Polymer processing applications, Thermoplastic sheet extrusion, Conjugate heat transfer problems, Very high-order accurate finite volume scheme, Method of manufactured solutions, High aspect ratio meshes

This chapter was adapted from R. Costa, J.M. Nóbrega, S. Clain, and G.J. Machado, Thermoplastic sheet extrusion cooling stage simulation with a very high-order accurate finite volume scheme, in preparation for Int. J. Heat Mass Tran. (2020)

4.1 Introduction

The ever-increasing complex manufacturing technologies in nowadays polymer processing industry, such as extrusion and injection moulding, demand strict control of many physical variables throughout the manufacturing process to ensure that the products meet the required specifications. Important thermodynamic phenomena, such as changes of physical state and heat transfer, occur in these manufacturing technologies and, therefore, the temperature distribution has a relevant importance in the process, among the other physical variables involved. For instance, a strictly controlled melting of the raw polymeric material is essential to avoid an excessively high temperature that might cause degradation and a permanent change of the chemical composition and the physical properties. On the other side, the cooling of the molten polymer is not less critical since the microstructure, and the physical properties of the final solidified product are mainly determined during this stage. Therefore, rigorous control of the temperature throughout the entire process in these manufacturing technologies is demanded to ensure the required quality of the products.

Constant cross-section plastic items, such as tubing, pipes, sheets, films, and also structural components, are usually manufactured by extrusion, which has a prominent presence in nowadays polymer processing industry. An illustration of a typical thermoplastic sheet extrusion line is provided in Figure 4.1. The process is continuous and consists of an extrusion machine composed of a screw rotating inside a heated barrel and gravity-fed with raw polymeric material from a top-mounted hopper placed in the rear [1–3]. The rotating screw compresses and forces the molten polymer forward through the barrel and several independently controlled heating units mounted in sequence gradually increase the temperature necessary for the melting. The viscous dissipation is another important thermodynamic phenomenon occurring inside the barrel, and the generated heat turns out more difficult to control the temperature and increases the risk of overheating. A die is mounted in the front of the barrel with openings corresponding to the desired product cross-section, through which the molten polymer, subjected to compression loads, is forced to flow. After leaving the die, the molten polymer is cooled and calibrated with specific cooling and calibration systems that ensure the required shape of the final solidified product with high precision. Finally, a caterpillar haul-off system is composed of puller rolls to provide continuous tension to the extruded profile, and a standard extrusion line ends with treating, cutting, winding, and storing equipment if needed.

The cooling stage in extrusion is crucial to ensure that the profile meets the required specifications, not only in terms of shape but also in terms of physical properties. Tubing and 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. On the other side, a stack of rotating metallic chilled rolls (also called calenders) is usually employed for sheets and films to avoid the drying difficulties associated with hydrophilic materials, which would be wet in water baths. Moreover, sheets and films are subject to a longitudinal stretch between the die opening and the rolls, such that thickness and width can be adjusted from controlling the rolls rotation speed [4]. Rolls are

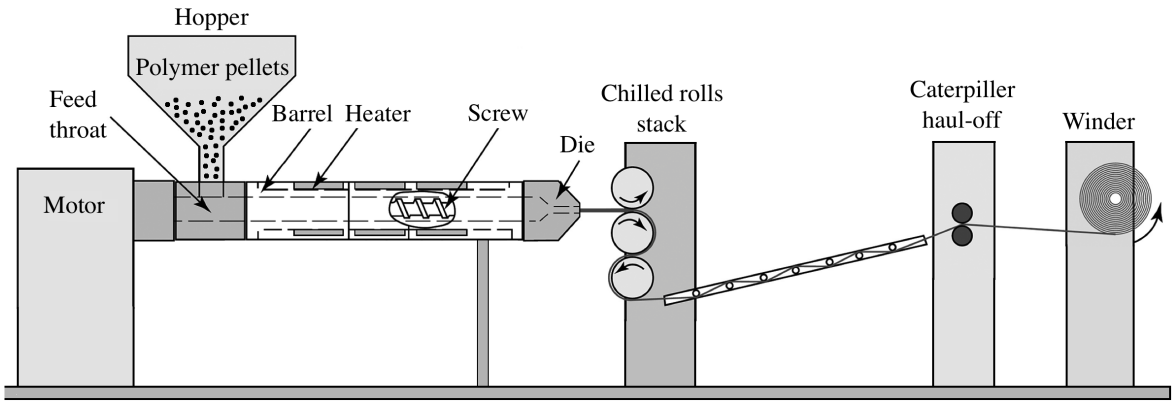


Figure 4.1: Typical thermoplastic sheet extrusion line (adapted from C. Rauwendaal, R. Gonzalez-Nunez, D. Rodrigue, Polymer processing: extrusion, in Encyclopedia of polymer science and technology, John Wiley & Sons (2017)).

typically hollowed, with water or oil circulating internally, being the temperature of each roll in the stack controlled separately to achieve the desired cooling rate gradually. Moreover, the surface of the rolls is usually highly polished to provide smoothness to the sheet and maximize the heat exchange with the roll. However, intentionally textured rolls are also employed to produce textured sheets. Contrarily to flat sheets and films, tubular films are extruded from a cylindrical die, which makes it impossible to use a stack of chilled rolls and, therefore, are usually cooled with only refrigerated air. Indeed, thinner profiles are more easily cooled (notice that sheets are usually considered having a thickness above 0.25 mm [5]). A more detailed representation of the thermoplastic sheet extrusion cooling stage, in a single chilled roll, is provided in Figure 4.2.

Although the cooling stage always aims at the solidification of the extruded sheet, the cooling rate has a significant influence on the physical properties of the final solidified product. The cooling rate strongly determines crystal growth and, consequently, the size of the crystallites present in the final solidified product, which affects the density, optical and barrier properties, coefficient of friction, impact behaviour, and other physical properties [5, 6]. A high cooling rate avoids larger crystallites and is usually desired to increase the surface smoothness and transparency of the sheet [4]. Moreover, it allows a higher extrusion speed increasing the productivity of the process and also avoids that insufficiently cooled sheets remain stuck to the roll and have release problems [7]. However, a high cooling rate creates higher internal residual stresses due to the drastic temperature difference between the colder sheet surface (in contact with the roll) and the hotter sheet surface (in contact with the surrounding air). Internal residual stresses are responsible for the shrinkage of the final solidified product and might lead to the separation between layers in multilayered sheets. In practical terms, the industry seeks to maximize the cooling rate that ensures an average temperature below the solidification point to prevent subsequent melting and, simultaneously, a sufficiently uniform temperature distribution to minimize internal residual stresses [8, 9]. Therefore, the cooling rate is usually the limiting factor of the sheet extrusion and great attention to these generally conflicting objectives is demanded, when aiming to optimize both the productivity and the quality of the

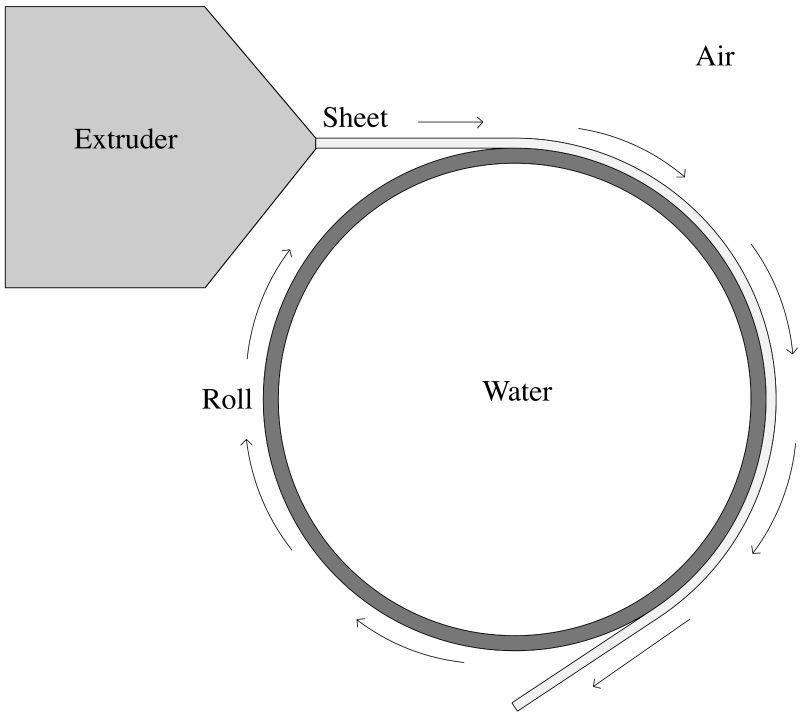


Figure 4.2: Schematic representation of the thermoplastic sheet extrusion cooling stage (the arrows represent the movement of the polymer sheet and the chilled roll).

process [8].

Several parameters are determinant for the cooling rate, namely the roll geometry (the rotation speed, diameter, angles (points) of contact, and thickness), the cooling conditions (the temperature and flow rate of the water inside the roll, the surrounding air temperature, and the natural or forced convection effects), the extrusion conditions (the speed, temperature, and thickness of the extruded sheet), and the metal and polymer physical properties (thermal conductivity, density, and specific heat capacity) [8]. The roll rotation speed, diameter, and angles of contact determine the effective time and length of contact (also called residence time) with the extruded sheet. On the other side, the thickness has a low influence since the thermal conductivity of the steel is significantly higher than the one of the polymer, meaning that heat is rapidly diffused. In that regard, the numerical simulation can be a valuable approach to investigate the complex heat exchanges of the process and perform optimized engineering design. The numerical approach enables an efficient determination of the influence and the relative importance of the several physical parameters involved in the cooling stage, which would be cumbersome, time-consuming, and expensive using the conventional fully experimental trial-and-error approaches. Moreover, the temperature of the sheet surface in contact with the roll is not fully accessible through experimental measurements, whereas the numerical approach does not have such limitations. A literature review retrieves some works regarding the application of numerical approaches in the optimization of the thermoplastic sheet extrusion [5, 10–12], however, substantial and comprehensive developments in the cooling stage do not exist up to the authors' knowledge.

The simulation of the thermoplastic sheet extrusion cooling stage consists in a non-isothermal process between materials with different physical properties and thermally coupled through a non-adiabatic contact, called conjugate heat transfer problem. Heat transfer governing equations are used to model the thermodynamics in each material, and the thermal coupling on the interface is achieved prescribing specific interface conditions. A generally adopted approach is to consider a perfect thermal contact between materials, for which the continuity of the temperature and the conservation of the normal conductive heat flux is prescribed. Perfect thermal contacts do not account for the effects of the surface roughness on the heat transfer through the interface and, therefore, are often used as convenient idealizations that provide valid results in a large variety of situations. On the contrary, the assumption of a perfect thermal contact is realistic in the case of the polymer sheet in contact with the roll as the former is still molten (and has no surface roughness) and the latter is intentionally highly polished. Moreover, the importance of the interface between thermally coupled materials is more pronounced as the problem scale decreases since the relative interface area, with resptect to the material volume, usually grows. Therefore, the effects of the conjugate heat transfer in the thermoplastic sheet extrusion cooling stage have a significant influence on the complex thermodynamics of the process.

The literature concerning discretization methods for conjugate heat transfer problems is vast and diverse, including spectral methods, finite difference methods, finite element methods, and finite volume methods, among others. Finite volume methods are a popular class of methods and particularly interesting in the context of thermodynamic problems due to the intrinsic conservation properties. The convergence order of the discretization method measures the rate at which the error of the computed approximate solution decreases under mesh refinement. Classical discretization methods provide at most a second-order of convergence, whereas very high-order of convergence methods are scarce due to the increased complexity and laborious implementation. However, a very high-order of convergence has shown to be a promising approach to solve complex problems and obtain approximate solutions with higher accuracy at a reduced computational cost. For instance, R. Costa et al., 2019 [13], proposed a very high-order accurate finite volume scheme capable of handling several types of thermal contacts, general boundary conditions, discontinuous coefficients, complex and curved geometries, and general polygonal meshes. Comprehensive numerical benchmarks prove that the proposed discretization method is very versatile, effectively achieves up to the sixth-order of convergence, and is more cost-effective in terms of accuracy than the classical second-order accurate methods.

The purpose of the present work is to develop a reliable and efficient numerical approach to investigate the complex heat exchanges involved in the thermoplastic sheet extrusion cooling stage. For that purpose, a steady-state two-dimensional conjugate heat transfer model with the appropriate boundary and interface conditions is proposed and solved with a very high-order accurate finite volume scheme to provide reliable approximate solutions and efficient simulations. The developed numerical approach is suitable for further parametric investigations aiming at assessing the influence of the

physical parameters involved in the thermoplastic sheet extrusion cooling stage on the cooling rate. Parametric investigations often require a large number of simulations and, therefore, the computational efficiency of each simulation is of crucial importance since it might severely impact the feasibility of the approach. In that regard, the very high-order of convergence is particularly interesting to reduce the computational cost without loss of accuracy.

The remaining sections of the chapter are organized as follows. Section 2 presents the conjugate heat transfer model, Section 3 recalls the very high-order accurate finite volume scheme, and Section 4 verifies the code and the discretization method in terms of accuracy and convergence order. Section 5 addresses the case study of the thermoplastic sheet extrusion cooling stage and the chapter is completed in Section 6 with the conclusions and some perspectives for future work.

4.2 Mathematical formulation

The thermoplastic sheet extrusion cooling stage consists in a non-isothermal and steady-state physical system involving materials with different physical properties and thermally coupled through a non-adiabatic contact. Therefore, a steady-state two-dimensional conjugate heat transfer model is considered for the problem comprising the sheet and the roll, whereas the heat transfer from the system to the surroundings (air and water) is taken into account with the appropriate boundary conditions.

Consider that the roll, the sheet, the water, the air, the inlet, and the outlet are identified with R, S, W, A, I, and O, respectively. The domain for the problem, denoted as Ω, is partitioned into the subdomain corresponding to the roll, denoted as Ω^{R}, and the subdomain corresponding to the sheet, denoted as Ω^{S} (see Figure 4.3). The boundaries of subdomain Ω^{R} are denoted as $\Gamma^{\mathrm{R,W}}$ and $\Gamma^{\mathrm{R,A}}$ and correspond to the surfaces of the roll in contact with water and air, respectively. The boundaries of subdomain Ω^{S} are denoted as $\Gamma^{\mathrm{S,I}}$, $\Gamma^{\mathrm{S,O}}$, and $\Gamma^{\mathrm{S,A}}$ and correspond to the inlet of the cooling stage, to the outlet of the cooling stage, and to the surfaces of the sheet in contact with air, respectively. The interface between subdomains Ω^{R} and Ω^{S} is denoted as $\Gamma^{\mathrm{R,S}}$ or $\Gamma^{\mathrm{S,R}}$ and corresponds to the contact between the roll and the sheet. Notice that although heat is exchanged with the water and the air, $\Gamma^{\mathrm{R,W}}$, $\Gamma^{\mathrm{R,A}}$, and $\Gamma^{\mathrm{S,A}}$ are boundaries of the problem. Such an assumption provides valid results prescribing the appropriate boundary conditions that take into account the effects of the natural and forced heat convection. Moreover, for any point $\boldsymbol{x}$ on boundaries $\Gamma^{\mathrm{R,W}}$, $\Gamma^{\mathrm{R,A}}$, $\Gamma^{\mathrm{S,I}}$, $\Gamma^{\mathrm{S,O}}$, and $\Gamma^{\mathrm{S,A}}$, vector functions $\boldsymbol{n}^{\mathrm{R,W}} := \boldsymbol{n}^{\mathrm{R,W}}(\boldsymbol{x})$, $\boldsymbol{n}^{\mathrm{R,A}} := \boldsymbol{n}^{\mathrm{R,A}}(\boldsymbol{x})$, $\boldsymbol{n}^{\mathrm{S,I}} := \boldsymbol{n}^{\mathrm{S,I}}(\boldsymbol{x})$, $\boldsymbol{n}^{\mathrm{S,O}} := \boldsymbol{n}^{\mathrm{S,O}}(\boldsymbol{x})$, and $\boldsymbol{n}^{\mathrm{S,A}} := \boldsymbol{n}^{\mathrm{S,A}}(\boldsymbol{x})$, respectively, provide the unit normal vectors outward to the associated subdomain, Ω^{R} or Ω^{S}. Similarly, for any point $\boldsymbol{x}$ on interface $\Gamma^{\mathrm{R,S}}$ (or $\Gamma^{\mathrm{S,R}}$), vector functions $\boldsymbol{n}^{\mathrm{R,S}} := \boldsymbol{n}^{\mathrm{R,S}}(\boldsymbol{x})$ and $\boldsymbol{n}^{\mathrm{S,R}} := \boldsymbol{n}^{\mathrm{S,R}}(\boldsymbol{x})$ provide the unit normal vectors outward to subdomains Ω^{R} and Ω^{S}, respectively. Notice that $\boldsymbol{n}^{\mathrm{R,S}}(\boldsymbol{x}) = -\boldsymbol{n}^{\mathrm{S,R}}(\boldsymbol{x})$ for any point $\boldsymbol{x}$ on interface $\Gamma^{\mathrm{R,S}}$.

Consider functions $T^{\mathrm{R}} := T^{\mathrm{R}}(\boldsymbol{x})$ and $T^{\mathrm{S}} := T^{\mathrm{S}}(\boldsymbol{x})$ standing for the temperature in subdomains

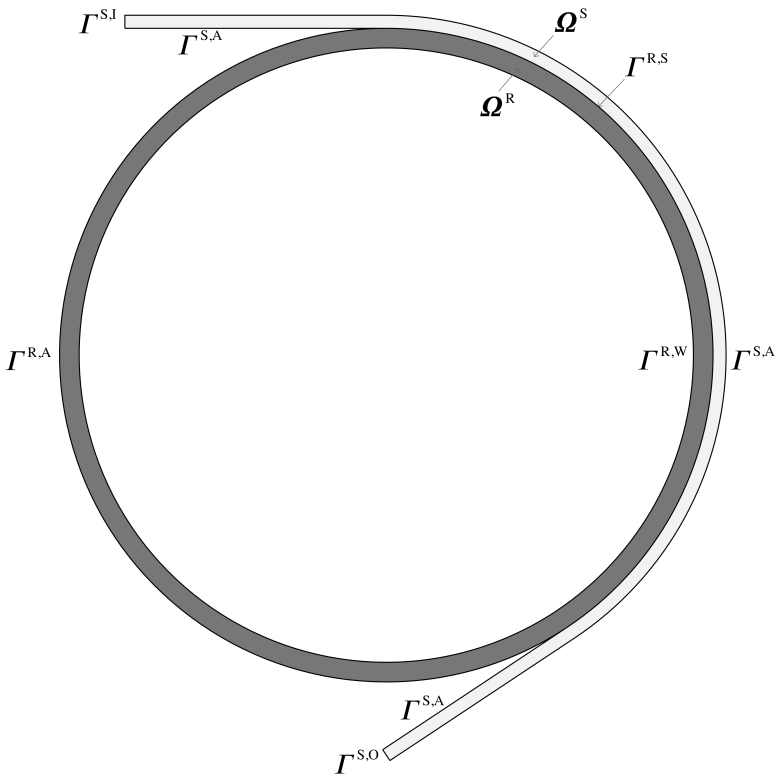


Figure 4.3: Domain, boundaries, and interface for the thermoplastic sheet extrusion cooling stage.

Ω^{R} and Ω^{S}, respectively. The conjugate heat transfer problem consists in seeking unknown temperature functions $T^{\mathrm{R}}(\boldsymbol{x})$ and $T^{\mathrm{S}}(\boldsymbol{x})$ in subdomains Ω^{R} and Ω^{S}, respectively, with governing partial differential equations given as

$$\nabla \cdot \left(\rho^{\mathrm{R}} c_{\mathrm{p}}^{\mathrm{R}} \boldsymbol{u}^{\mathrm{R}} T^{\mathrm{R}} - \kappa^{\mathrm{R}} \nabla T^{\mathrm{R}} \right) = 0, \quad \text{in } \Omega^{\mathrm{R}}, \tag{4.1}$$

$$\nabla \cdot \left(\rho^{\mathrm{S}} c_{\mathrm{p}}^{\mathrm{S}} \boldsymbol{u}^{\mathrm{S}} T^{\mathrm{S}} - \kappa^{\mathrm{S}} \nabla T^{\mathrm{S}} \right) = 0, \quad \text{in } \Omega^{\mathrm{S}}, \tag{4.2}$$

where, in subdomain Ω^{P}, coefficients ρ^{P}, c_{p}^{P}, and κ^{P} are the density, the specific heat capacity, and the thermal conductivity, respectively, where P stands both for the roll ($P = \mathrm{R}$) and the sheet ($P = \mathrm{S}$). Moreover, vector function $\boldsymbol{u}^{P} := \boldsymbol{u}^{P}(\boldsymbol{x})$ provide the velocity vectors in subdomain Ω^{P}, $P \in \{\mathrm{R}, \mathrm{S}\}$.

The model is complemented with Dirichlet and Neumann boundary conditions, as follows. A constant Dirichlet boundary condition is prescribed on boundary subset $\Gamma^{\mathrm{S,I}}$, to impose a uniform temperature on the inlet of the domain. A homogeneous Neumann boundary condition is prescribed on boundary subset $\Gamma^{\mathrm{S,O}}$, to impose a vanishing temperature normal derivative on the outlet of the domain. A Neumann boundary condition as a function of the temperature is prescribed on boundary subset $\Gamma^{\mathrm{R,W}}$, to impose forced heat convection on the inner roll surface in contact with water. Similarly, Neumann boundary conditions as a function of the temperature are prescribed on boundaries $\Gamma^{\mathrm{R,A}}$ and $\Gamma^{\mathrm{S,A}}$, to impose natural heat convection on the external surface of the roll and the sheet in contact

with air. The case study boundary conditions are given as

$$T^{\mathrm{S}} = T^{\mathrm{I}}, \qquad \text{on } \Gamma^{\mathrm{S,I}}, \tag{4.3}$$

$$-\kappa^{\mathrm{S}} \nabla T^{\mathrm{S}} \cdot \boldsymbol{n}^{\mathrm{S,O}} = 0, \qquad \text{on } \Gamma^{\mathrm{S,O}}, \tag{4.4}$$

$$-\kappa^{\mathrm{R}} \nabla T^{\mathrm{R}} \cdot \boldsymbol{n}^{\mathrm{R,W}} = h^{\mathrm{R,W}} \left(T^{\mathrm{R}} - T_{\infty}^{\mathrm{A}} \right), \qquad \text{on } \Gamma^{\mathrm{R,W}}, \tag{4.5}$$

$$-\kappa^{\mathrm{R}} \nabla T^{\mathrm{R}} \cdot \boldsymbol{n}^{\mathrm{R,A}} = h^{\mathrm{R,A}} \left(T^{\mathrm{R}} - T_{\infty}^{\mathrm{W}} \right), \qquad \text{on } \Gamma^{\mathrm{R,A}}, \tag{4.6}$$

$$-\kappa^{\mathrm{S}} \nabla T^{\mathrm{S}} \cdot \boldsymbol{n}^{\mathrm{S,A}} = h^{\mathrm{S,A}} \left(T^{\mathrm{S}} - T_{\infty}^{\mathrm{A}} \right), \qquad \text{on } \Gamma^{\mathrm{S,A}}, \tag{4.7}$$

where T^{I} is the temperature of the sheet immediately after leaving the die, T_{∞}^{W} and T_{∞}^{A} are the temperature of the water and the air, respectively, and $h^{\mathrm{R,W}}$, $h^{\mathrm{R,A}}$, and $h^{\mathrm{S,A}}$ are the heat transfer coefficients between the roll and the water, between the roll and the air, and between the sheet and the air, respectively.

The model is also complemented with the continuity interface conditions (also called homogeneous jump interface conditions) on interface $\Gamma^{\mathrm{R,S}}$ to impose the continuity of the temperature and the conservation of the normal conductive heat flux, given as

$$T^{\mathrm{R}} = T^{\mathrm{S}}, \qquad \text{on } \Gamma^{\mathrm{R,S}}, \tag{4.8}$$

$$-\kappa^{\mathrm{R}} \nabla T^{\mathrm{R}} \cdot \boldsymbol{n}^{\mathrm{R,S}} - \kappa^{\mathrm{S}} \nabla T^{\mathrm{S}} \cdot \boldsymbol{n}^{\mathrm{S,R}} = 0, \qquad \text{on } \Gamma^{\mathrm{R,S}}. \tag{4.9}$$

Notice that the continuity interface conditions assume a perfect thermal contact between the surfaces of the roll and the sheet, which is appropriate for the problem since both surfaces move at the same speed an the roughness effects are negligible (other interface conditions should be imposed in the case of imperfect thermal contacts).

4.3 Numerical discretization

To solve the conjugate heat transfer problem, general polygonal meshes $\mathcal{M}^{\mathrm{R}}$ and $\mathcal{M}^{\mathrm{S}}$ are generated for subdomains Ω^{R} and Ω^{S}, respectively, and consist of n^{R} and n^{S} cells, respectively. Cells are denoted as c_i with index $i \in \{1, \ldots, n^{\mathrm{R}} + n^{\mathrm{S}}\}$, whereas edges between cell c_i and another cell c_j with index $j \in \{1, \ldots, n^{\mathrm{R}} + n^{\mathrm{S}}\} \setminus \{i\}$ are denoted as e_{ij} (in the case of a boundary edge, index j is replaced with F). Moreover, for edge e_{ij}, $|e_{ij}|$ is the associated length, $\boldsymbol{s}_{ij}$ is the associated unit normal vector outward to cell c_i, and $\boldsymbol{q}_{ij,r}$, $r = 1, \ldots, R$ are the associated quadrature points for an R-points quadrature rule.

The discretization of the conjugate heat transfer problem follows the proposed method in R. Costa et al., 2019 [13], which is shortly recalled herein. A partitioning method decomposes the conjugate heat transfer problem into individual subproblems of heat transfer in each subdomain and

converts the interface conditions into boundary conditions. On the interface, a Dirichlet boundary condition imposes the temperature in one subproblem, and a Neumann boundary condition imposes the normal conductive heat flux in the other. The thermal coupling between the subproblems consists in providing the appropriate temperature and normal conductive heat flux determined from the solution in the adjacent subdomain. Given that, the solution of the partitioned problem satisfies the interface conditions and, therefore, corresponds to the solution of the conjugate heat transfer problem. Moreover, since each subproblem consists in a governing equation with only boundary conditions, the discretization is performed separately in each subdomain with the cost of an additional, but straightforward and efficient, numerical treatment for the thermal coupling.

The finite volume method is considered for the discretization of the subproblems and consists in applying the divergence theorem to the integral form of Equations (4.1) and (4.2) in each cell c_i of mesh $\mathcal{M}^P$, given as

$$\int_{\partial c_i} \left(\rho^P c_{\mathrm{p}}^P \boldsymbol{u}^P T^P - \kappa^P \nabla T^P \right) \cdot \boldsymbol{s}_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 0, \tag{4.10}$$

for $P \in \{\mathrm{R}, \mathrm{S}\}$, where ∂c_i stands for the cell boundary and $\boldsymbol{s}_i(\boldsymbol{x})$ is the associated outward unit normal vector. Consider index set $\mathcal{N}_i$ gathering indices $j \in \{1, \ldots, n^{\mathrm{R}} + n^{\mathrm{S}}\} \cup \{F\} \setminus \{i\}$ such that edge e_{ij} belongs to cell c_i. Then, Equation (4.10) is rewritten in the discrete form in terms of numerical integrals determined with an R-points quadrature rule with weights ζ_r, $r = 1, \ldots, R$, given as

$$\sum_{j \in \mathcal{N}_i} |e_{ij}| \left[\sum_{r=1}^{R} \zeta_r \left(C_{ij,r} + D_{ij,r} \right) \right] = \mathcal{O}(h_i^{\alpha}), \tag{4.11}$$

where $h_i = |c_i|^{1/2}$, α is the convergence order of the quadrature rule, and $C_{ij,r}$ and $D_{ij,r}$ stand for the exact convective and conductive heat fluxes, respectively, at quadrature point $\boldsymbol{q}_{ij,r}$ of edge e_{ij}, given as

$$C_{ij,r} = \left(\rho^P c_{\mathrm{p}}^P \boldsymbol{u}^P \left(\boldsymbol{q}_{ij,r} \right) \cdot \boldsymbol{s}_{ij} \right) T^P \left(\boldsymbol{q}_{ij,r} \right), \tag{4.12}$$

$$D_{ij,r} = -\kappa^P \nabla T^P \left(\boldsymbol{q}_{ij,r} \right) \cdot \boldsymbol{s}_{ij}. \tag{4.13}$$

The numerical approximations to exact convective and conductive heat fluxes $C_{ij,r}$ and $D_{ij,r}$, respectively, are computed with the very high-order finite volume scheme proposed in R. Costa et al., 2019 [13]. The scheme consists in determining the values and gradients of temperature function $T^P(\boldsymbol{x})$ in Equations (4.12) and (4.13) from local polynomial reconstructions of degree d. An optimal $(d+1)$th-order of convergence is expected using a polynomial degree d and, therefore, the scheme is capable of achieving both the classical second-order of convergence (for degree $d = 1$) and a very high-order of convergence (for degrees $d > 1$).

For each cell c_i of mesh $\mathcal{M}^P$, consider T_i as the cell mean-value of temperature function

$T^P(\boldsymbol{x})$, given as $T_i = (1/|c_i|)\int_{c_i} T^P(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, where $|c_i|$ is the cell area. Then, for a given mesh element, the polynomial reconstruction method consists in approximating, in the least-squares sense, the cell mean-values in the vicinity to a polynomial function with coefficients to determine. Moreover, for the mesh elements on the boundaries, the polynomial reconstruction method takes into account the prescribed boundary conditions considering specific linear constraints in the associated least-squares method. For each inner edge e_{ij}, boundary edge e_{iF}, and cell c_i, the associated polynomial reconstructions are denoted as $\widetilde{T}_{ij}(\boldsymbol{x})$, $\widehat{T}_{iF}(\boldsymbol{x})$, and $\widehat{T}_i(\boldsymbol{x})$, respectively.

Consider the convention $[a]^+ = \max(a,0)$ and $[a]^- = \min(a,0)$ as an upwind indicator. Then, the numerical convective and conductive heat fluxes are given as follows:

- For each inner edge e_{ij} of mesh $\mathcal{M}^P$, numerical convective and conductive heat fluxes, denoted as $\mathcal{C}_{ij,r}$ and $\mathcal{D}_{ij,r}$, respectively, at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1,\ldots,R$, are given as

$$\mathcal{C}_{ij,r} = \left[\rho^P c_{\mathrm{p}}^P \boldsymbol{u}^P\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}\right]^+ \widehat{T}_i\left(\boldsymbol{q}_{ij,r}\right) + \left[\rho^P c_{\mathrm{p}}^P \boldsymbol{u}^P\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}\right]^- \widehat{T}_j\left(\boldsymbol{q}_{ij,r}\right), \tag{4.14}$$

$$\mathcal{D}_{ij,r} = -\kappa^P \nabla\widetilde{T}_{ij}\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}. \tag{4.15}$$

- For each boundary edge e_{iF} of mesh $\mathcal{M}^P$, numerical convective and conductive heat fluxes, denoted as $\mathcal{C}_{iF,r}$ and $\mathcal{D}_{iF,r}$, respectively, at quadrature points $\boldsymbol{q}_{iF,r}$, $r = 1,\ldots,R$, are given as

$$\mathcal{C}_{iF,r} = \left[\rho^P c_{\mathrm{p}}^P \boldsymbol{u}^P\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}\right]^+ \widehat{T}_i\left(\boldsymbol{q}_{iF,r}\right) + \left[\rho^P c_{\mathrm{p}}^P \boldsymbol{u}^P\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}\right]^- \widehat{T}_{iF}\left(\boldsymbol{q}_{iF,r}\right), \tag{4.16}$$

$$\mathcal{D}_{iF,r} = -\kappa^P \nabla\widehat{T}_{iF}\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}. \tag{4.17}$$

Replacing exact convective and conductive heat fluxes $C_{iF,r}$ and $D_{iF,r}$ in Equation (4.11) with corresponding numerical approximations $\mathcal{C}_{iF,r}$ and $\mathcal{D}_{iF,r}$, respectively, an explicit system of linear equations is assembled with one equation per cell. The variables to determine in the system of linear equations correspond to approximations of the cell mean-values of temperature function $T^P(\boldsymbol{x})$ in mesh $\mathcal{M}^P$, $P \in \{\mathrm{R},\mathrm{S}\}$. Therefore, the associated number of degrees of freedom to determine is $DOF = n^{\mathrm{R}} + n^{\mathrm{S}}$. The system of linear equations derived from the discretization method is solved with a preconditioned version of the GMRES method, and the stopping criterion corresponds to a residual tolerance.

4.4 Code verification

The proposed discretization method to solve conjugate heat transfer problems is implemented in an in-house library written in Fortran, and verification of the code is performed in terms of accuracy, convergence order, stability, robustness, and execution time. The method of verification is based on manufactured solutions and consists in addressing a specific case of the conjugate heat transfer

problem having an analytic solution that satisfies the governing equations and the prescribed boundary and interface conditions. Simulations with successive finer meshes generated for the domain are carried out, and the computed approximate solutions are compared with the analytic solution providing the accuracy and the convergence orders under mesh refinement of the proposed discretization method.

The solution of the system of linear equations derived from the discretization method provides approximations of the cell mean-value of temperature functions $T^P(\boldsymbol{x})$ in meshes $\mathcal{M}^P$, $P \in \{\mathrm{R}, \mathrm{S}\}$. Consider that for each cell c_i of mesh $\mathcal{M}^P$, the associated approximate cell mean-values is denoted as T_i^* and the associated exact cell mean-value is denoted $\overline{T}_i$ and given as

$$\overline{T}_i = \frac{1}{|c_i|} \int_{c_i} T^P(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{4.18}$$

Then, the errors in the L^1- and L^∞-norm, denoted as E_1 and E_∞, respectively, are determined as

$$E_1 = \frac{\sum_{i=1}^{DOF} \left| T_i^* - \overline{T}_i \right| |c_i|}{\sum_{i=1}^{DOF} |c_i|}, \tag{4.19}$$

$$E_\infty = \max_{i=1}^{DOF} \left| T_i^* - \overline{T}_i \right|. \tag{4.20}$$

Consider two consecutively finer meshes with number of degrees of freedom DOF_1 and DOF_2 and the computed approximate solutions provide errors $E_{1,1}$ and $E_{1,2}$, respectively, in the L^1-norm and errors $E_{\infty,1}$ and $E_{\infty,2}$, respectively, in the L^∞-norm. Then, the convergence orders for the errors in the L^1- and L^∞-norms, denoted as O_1 and O_∞, respectively, are given as

$$O_1 = 2 \left| \frac{\ln\left(E_{1,1}/E_{1,2}\right)}{\ln(DOF_1/DOF_2)} \right|, \tag{4.21}$$

$$O_\infty = 2 \left| \frac{\ln\left(E_{\infty,1}/E_{\infty,2}\right)}{\ln(DOF_1/DOF_2)} \right|. \tag{4.22}$$

The proposed discretization method is applied to the conjugate heat transfer problem considering polynomial degrees $d = 1, 3$ to verify whether the expected optimal second- and fourth-orders of convergence are effectively achieved. Besides the errors and convergence orders, the execution time of the simulations is also reported, denoted as T_S, in seconds.

The addressed test case consists of an annular section centred at the origin with an amplitude of $\alpha = \pi/4$ replicating the thermal contact between the roll and the sheet (see Figure 4.4). The inner layer of the geometry represents a roll section with a thickness of $t^\mathrm{R} = 3$ mm, and corresponds to subdomain Ω^R. In contrast, the outer layer represents a sheet section with a thickness of $t^\mathrm{S} = 2$ mm,

and corresponds to subdomain Ω^{S}. The internal boundary of the domain has a radius of $r_{\mathrm{I}} = 97$ mm, the external boundary has a radius of $r_{\mathrm{E}} = 102$ mm, and the interface separating the roll and the sheet has a radius of $r_{\mathrm{M}} = 100$ mm.

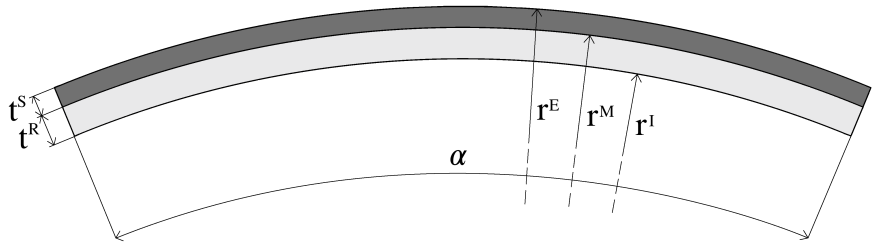


Figure 4.4: Geometry for the annular section test case.

The same angular speed of $\omega^{\mathrm{R}} = \omega^{\mathrm{S}} = 0.1$ rad/s is considered for both the roll and the sheet and, therefore, there is no discontinuity in the corresponding linear velocity provided with vector functions $\boldsymbol{u}^{\mathrm{R}}(\boldsymbol{x})$ and $\boldsymbol{u}^{\mathrm{S}}(\boldsymbol{x})$. The physical properties for the roll and the sheet are assumed those corresponding to the steel and the polystyrene, respectively, as follows:

- Roll: thermal conductivity of $\kappa^{\mathrm{R}} = 30$ W/(m·K), density of $\rho^{\mathrm{R}} = 7700$ kg/m^3, and specific heat capacity of $c_{\mathrm{p}}^{\mathrm{R}} = 500$ J/(K·kg).
- Sheet: thermal conductivity of $\kappa^{\mathrm{S}} = 0.2$ W/(m·K), density of $\rho^{\mathrm{S}} = 900$ kg/m^3, and specific heat capacity of $c_{\mathrm{p}}^{\mathrm{S}} = 1300$ J/(K·kg).

The analytic solutions assigned to this test case (see Figure 4.5) are expressed in polar coordinates (r, θ), with radius $r^2 = x^2 + y^2$ and polar angle $\theta = \arctan(y/x)$, given as

$$\phi^P(r) = a^P \ln(r) + b^P, \quad \text{in } \Omega^P, \tag{4.23}$$

for $P \in \{\mathrm{R}, \mathrm{S}\}$, with parameters $a^{\mathrm{R}}, b^{\mathrm{R}}, a^{\mathrm{S}}, b^{\mathrm{S}} \in \mathbb{R}$. Notice that, although the analytic solutions are provided in polar coordinates, the problem is numerically solved in Cartesian coordinates. Contrarily to the general case in the method of manufactured solutions, no artificial source terms are required in the governing equations with these analytic solutions.

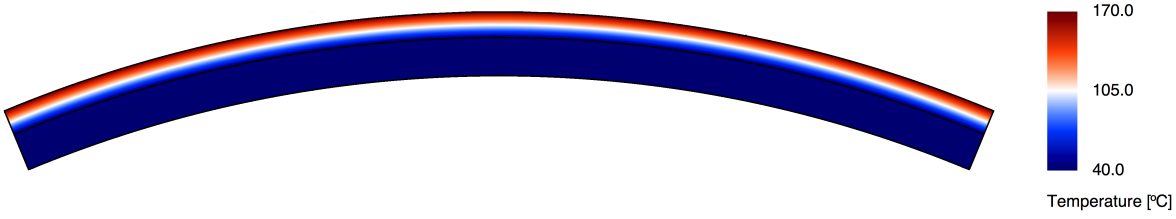


Figure 4.5: Analytic solution for the annular section test case.

On the boundaries of each subdomain, specific boundary conditions are assigned to satisfy the corresponding analytic solutions, as follows:

- A Dirichlet boundary condition is prescribed on the left boundaries of both subdomains, corresponding to points with an angle of $\theta = -\pi/8$, given as

$$T^{\mathrm{P}} = a^P \log(r) + b^P, \quad \text{on } \theta = -\pi/8. \tag{4.24}$$

- A Homogeneous Neumann boundary condition is prescribed on the right boundaries of both subdomains, corresponding to points with an angle of $\theta = \pi/8$, given as

$$-\kappa^P \nabla T^P \cdot \boldsymbol{n}^P = 0, \quad \text{on } \theta = \pi/8, \tag{4.25}$$

  where vector functions $\boldsymbol{n}^P := \boldsymbol{n}^P(\boldsymbol{x})$ provide the unit normal vectors outward to associated subdomain Ω^P.

- A Neumann boundary condition as a function of the temperature is prescribed on the internal boundary of subdomain Ω^{R}, corresponding to points with a radius of $r = r_{\mathrm{I}}$, given as

$$-\kappa^{\mathrm{R}} \nabla T^{\mathrm{R}} \cdot \boldsymbol{n}^{\mathrm{R}} = h^{\mathrm{R,W}} \left(T^{\mathrm{R}} - T_\infty^{\mathrm{W}} \right), \quad \text{on } r = r_{\mathrm{I}}, \tag{4.26}$$

  assuming a water temperature of $T_\infty^{\mathrm{W}} = 30\ ^\circ\mathrm{C}$.

- A Neumann boundary condition as a function of the temperature is prescribed on the external boundary of subdomain Ω^{S}, corresponding to points with a radius of $r = r_{\mathrm{E}}$, given as

$$-\kappa^{\mathrm{S}} \nabla T^{\mathrm{S}} \cdot \boldsymbol{n}^{\mathrm{S}} = h^{\mathrm{S,A}} \left(T^{\mathrm{S}} - T_\infty^{\mathrm{A}} \right), \quad \text{on } r = r_{\mathrm{E}}, \tag{4.27}$$

  assuming an air temperature of $T_\infty^{\mathrm{A}} = 20\ ^\circ\mathrm{C}$.

The prescribed Neumann boundary conditions as a function of the temperature take into account the effects of the forced and natural convection due to the water and air circulation. A specific temperature is assumed on the internal and external boundaries such that the associated heat transfer coefficients are determined, and the prescribed Neumann boundary conditions satisfy the analytic solutions. The temperature of the internal and external boundaries are assumed of $T^{\mathrm{R,W}} = 40\ ^\circ\mathrm{C}$ and $T^{\mathrm{S,A}} = 170\ ^\circ\mathrm{C}$, respectively, for which the associated heat transfer coefficients are determined as $h^{\mathrm{R,W}} = \kappa^{\mathrm{R}} a^{\mathrm{R}} / r_{\mathrm{I}} \left(T^{\mathrm{R,W}} - T_\infty^{\mathrm{W}} \right)$ W/(K·m^2) and $h^{\mathrm{S,A}} = -\kappa^{\mathrm{S}} a^{\mathrm{S}} / r_{\mathrm{E}} \left(T^{\mathrm{S,A}} - T_\infty^{\mathrm{A}} \right)$ W/(K·m^2), respectively.

On the interface, the continuity interface conditions are prescribed and parameters a^{R}, a^{S}, b^{R}, and b^{S} in the analytic solutions are determined such that the boundary conditions and the continuity

interface conditions are simultaneously satisfied, and are determined as

$$a^{\mathrm{R}} = -c\kappa^{\mathrm{S}}\left(T^{\mathrm{R,W}} - T^{\mathrm{S,A}}\right), \tag{4.28}$$

$$a^{\mathrm{S}} = -c\kappa^{\mathrm{R}}\left(T^{\mathrm{R,W}} - T^{\mathrm{S,A}}\right), \tag{4.29}$$

$$b^{\mathrm{R}} = c\left(T^{\mathrm{R,W}}\kappa^{\mathrm{R}}\ln\left(\frac{r_{\mathrm{M}}}{r_{\mathrm{E}}}\right) - T^{\mathrm{R,W}}\kappa^{\mathrm{S}}\ln(r_{\mathrm{M}}) + T^{\mathrm{S,A}}\kappa^{\mathrm{S}}\ln(r_{\mathrm{I}})\right), \tag{4.30}$$

$$b^{\mathrm{S}} = -c\left(T^{\mathrm{S,A}}\kappa^{\mathrm{S}}\ln\left(\frac{r_{\mathrm{M}}}{r_{\mathrm{I}}}\right) - T^{\mathrm{S,A}}\kappa^{\mathrm{R}}\ln(r_{\mathrm{M}}) + T^{\mathrm{R,W}}\kappa^{\mathrm{R}}\ln(r_{\mathrm{E}})\right), \tag{4.31}$$

$$c = \frac{1}{\kappa^{\mathrm{R}}\ln\left(\frac{r_{\mathrm{M}}}{r_{\mathrm{E}}}\right) + \kappa^{\mathrm{S}}\ln\left(\frac{r_{\mathrm{I}}}{r_{\mathrm{M}}}\right)}. \tag{4.32}$$

Successively finer uniform structured quadrilateral meshes (with no curved elements) with aspect ratios of $AR = 1, 2, 5, 10, 20$ are generated for each subdomain. The aspect ratio refers to the outer cells in the subdomain corresponding to the sheet, whereas the size of the remaining cells is adapted accordingly to obtain a structured mesh. Having meshes with unitary aspect ratio, successively finer meshes with higher aspect ratios are generated merging cells proportionally in the angular direction (see Figures 4.6 to 4.10). For each aspect ratio, four successively finer meshes are generated with the coarsest denoted as M1, the coarsest medium as M2, the finest medium as M3, and the finest as M4, having the number of cells presented in Table 4.1. Higher aspect ratios are a promising approach to reduce the computational effort of the simulations when the solution varies mostly in one specific direction or tangentially to a boundary or interface (as in the case of boundary layers). Indeed, the analytic solutions assigned to this test case are invariant in the angular direction, and similar behaviour is expected in the thermoplastic sheet extrusion cooling stage as a limit situation for increasing extrusion speeds. Therefore, comparing the results between meshes with different aspect ratios, the influence of the aspect ratio on the discretization method is assessed in terms of accuracy, convergence order, and execution time.

Table 4.1: Number of cells in the structured quadrilateral meshes for the annular section test case.

Mesh	$AR = 1$	$AR = 2$	$AR = 5$	$AR = 10$	$AR = 20$
M1	10 x 200	10 x 100	10 x 40	10 x 20	10 x 10
M2	20 x 400	20 x 200	20 x 80	20 x 40	20 x 20
M3	40 x 800	40 x 400	40 x 160	40 x 80	40 x 40
M4	80 x 1 600	80 x 800	80 x 320	80 x 160	80 x 80

The errors and convergence orders are reported in Table 4.2. As observed, the second- and fourth-orders of convergence are effectively achieved, applying the discretization method with polynomial degrees $d = 1, 3$, respectively. The convergence orders follow the results obtained in

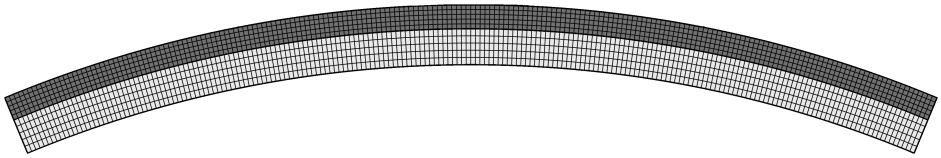


Figure 4.6: Coarsest quadrilateral mesh with $AR = 1$ for the annular section test case.

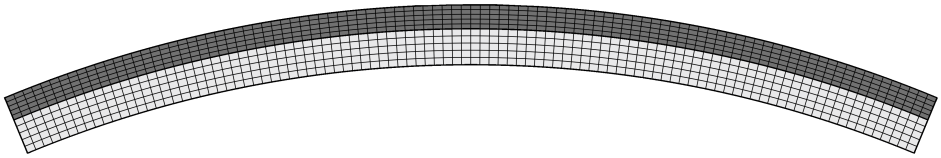


Figure 4.7: Coarsest quadrilateral mesh with $AR = 2$ for the annular section test case.

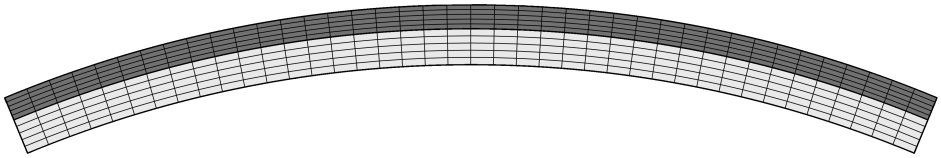


Figure 4.8: Coarsest quadrilateral mesh with $AR = 5$ for the annular section test case.

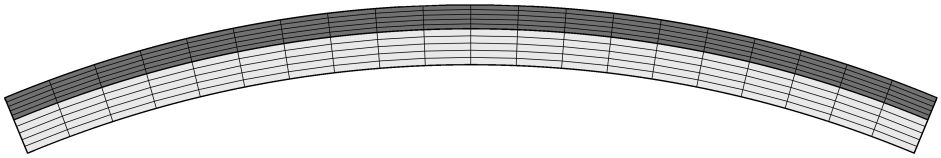


Figure 4.9: Coarsest quadrilateral mesh with $AR = 10$ for the annular section test case.

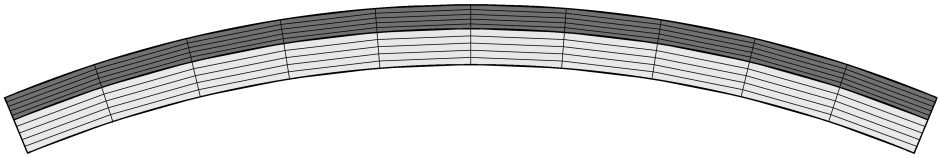


Figure 4.10: Coarsest quadrilateral mesh with $AR = 20$ for the annular section test case.

previous numerical benchmarks, and the discretization method converges optimally, both in terms of the L^1- and L^∞-norms. Moreover, the ratio between the errors in the L^∞- and L^1-norms obtained from the same simulation has, in general, a magnitude order in the tens. Such a ratio typically indicates the absence of spurious solutions, and the error distribution is relatively uniform in the domain.

The optimal converge orders are always achieved regardless of the mesh. However, accuracy deterioration with increasing aspect ratio is observed, both in terms of the L^1- and L^∞-norms, being more pronounced in the case of the fourth-order of convergence. Notice that the assigned analytic solutions to this test case are invariant in the angular direction and, therefore, the same accuracy would be expected in an ideal situation. However, the discretization method relies on local polynomial reconstructions approximating the cell mean-values in the vicinity, which are computed from the

Table 4.2: Errors, convergence orders, and execution time obtained in the annular section test case.

	Second-order of convergence					Fourth-order of convergence				
Mesh	E_1	O_1	E_∞	O_∞	T_S [s]	E_1	O_1	E_∞	O_∞	T_S [s]
$AR = 1$										
M1	8.60E−03	—	5.68E−02	—	0.10	8.64E−07	—	1.41E−05	—	0.10
M2	2.08E−03	2.05	1.25E−02	2.19	1.49	5.62E−08	3.94	8.47E−07	4.06	2.45
M3	7.86E−04	1.40	4.69E−03	1.41	48.56	5.11E−09	3.46	4.90E−08	4.11	112.56
M4	1.37E−04	2.52	1.09E−03	2.10	662.59	5.97E−10	3.10	5.78E−09	3.08	1 295.09
$AR = 2$										
M1	1.58E−02	—	1.60E−01	—	0.03	5.35E−06	—	1.87E−04	—	0.03
M2	2.93E−03	2.44	4.01E−02	2.00	0.23	2.13E−07	4.65	6.71E−06	4.80	0.40
M3	5.60E−04	2.39	1.02E−02	1.98	3.36	2.20E−08	3.27	3.72E−07	4.17	13.68
M4	1.18E−04	2.25	2.61E−03	1.96	129.07	3.30E−09	2.74	3.76E−08	3.31	306.09
$AR = 5$										
M1	7.06E−02	—	1.01E+00	—	<0.01	2.64E−04	—	6.36E−03	—	0.01
M2	1.62E−02	2.13	2.54E−01	1.99	0.02	4.78E−06	5.79	1.51E−04	5.40	0.03
M3	3.70E−03	2.13	6.28E−02	2.01	0.50	1.97E−07	4.61	8.77E−06	4.11	1.13
M4	7.96E−04	2.22	1.55E−02	2.01	27.95	1.49E−08	3.72	4.97E−07	4.14	61.59
$AR = 10$										
M1	2.78E−01	—	4.20E+00	—	<0.01	5.57E−03	—	1.14E−01	—	<0.01
M2	4.05E−02	2.78	1.05E+00	2.00	0.02	1.28E−04	5.44	2.51E−03	5.50	0.02
M3	7.07E−03	2.52	2.50E−01	2.07	0.18	4.43E−06	4.86	1.51E−04	4.06	0.26
M4	2.46E−03	1.52	5.93E−02	2.07	4.76	1.58E−07	4.81	8.69E−06	4.12	11.55
$AR = 20$										
M1	1.56E+00	—	1.85E+01	—	<0.01	1.34E−01	—	1.96E+00	—	<0.01
M2	2.07E−01	2.91	4.38E+00	2.08	0.01	2.34E−03	5.84	3.98E−02	5.63	0.01
M3	3.86E−02	2.42	1.01E+00	2.12	0.06	9.37E−05	4.64	2.42E−03	4.04	0.07
M4	6.54E−03	2.56	2.32E−01	2.12	1.42	3.91E−06	4.58	1.41E−04	4.10	1.81

least-squares method. In turn, the reconstruction matrices in the least-squares method are mainly geometric and, therefore, the associated condition numbers strongly depend on the structure of the mesh. Indeed, the work of A. Jalali et al., 2013 [14], verified that the condition numbers of the reconstruction matrices increase with increasing aspect ratio of the mesh and is asymptotically larger for higher convergence orders. As a consequence, the discretization method suffers from deterioration in terms of accuracy and stability.

The number of GMRES iterations for the previous simulations is reported in Table 4.3. As observed, the number of GMRES iterations and the execution time generally increase with increasing convergence order of the discretization method. In fact, although polynomial reconstructions with higher polynomial degrees provide more accuracy, denser coefficients matrices also results from the larger stencils required for the least-squares method, having higher condition numbers. Consequently, more GMRES iterations are needed to achieve the same residual tolerance, which, in turn, increase the execution time of the simulations. On the other side, the number of GMRES iterations and the execution time also increase with increasing aspect ratio of the mesh, comparing meshes with the same number of cells. Moreover, such behaviour is observed comparing the results for the same polynomial degree. Therefore, the higher condition number of the coefficients matrices is, again, a possible explanation for such behaviour, but, in that case, due to the high disparity in the propagation of information along each dimension.

Table 4.3: Number of GMRES iterations obtained in the annular section test.

	Second-order of convergence					Fourth-order of convergence				
Mesh	$AR = 1$	$AR = 2$	$AR = 5$	$AR = 10$	$AR = 20$	$AR = 1$	$AR = 2$	$AR = 5$	$AR = 10$	$AR = 20$
M1	179	88	39	31	21	188	97	64	48	43
M2	441	218	88	77	56	493	245	123	90	74
M3	3 415	464	257	184	168	6 024	1 481	353	225	172
M4	10 452	4 626	2 476	871	477	16 971	8 498	4 377	1 513	468

The execution time of the simulations are also reported in Table 4.2, whereas Figures 4.11 and 4.12 provides a plot of the errors in the L^1- and L^∞-norms as a function of the execution time. Regardless of the aspect ratio of the mesh, the fourth-order of convergence always performs more efficiently than the second-order of convergence and provides the same level of accuracy in significantly less execution time. Therefore, the results support that higher-orders of convergence provide more efficient trade-offs between accuracy and execution time. For instance, considering $AR = 1$, to achieve an error of $E_1 = 10^{-6}$ °C in the L^1-norm, the second- and fourth-orders of convergence require a execution time of approximately $T_\mathrm{S} = 100,000$ s and $T_\mathrm{S} = 0.1$ s, respectively. The second-order of convergence does not achieve such accuracy within the meshes considered for this test case and, therefore, the indicated execution time is a prediction from the extrapolation of the obtained error convergence. In any case, the second-order of convergence requires one day of simulation in comparison with the minimal execution time, below one second, taken by the fourth-order of convergence.

Higher aspect ratios deteriorate the accuracy of the approximate solutions for the reasons already presented. However, they also reduce the number of degrees of freedom to determine and, ultimately, the execution time. Given that, there is a trade-off between accuracy and execution time, which depends on the aspect ratio of the mesh. From the results obtained, the trade-off seems to deteriorate

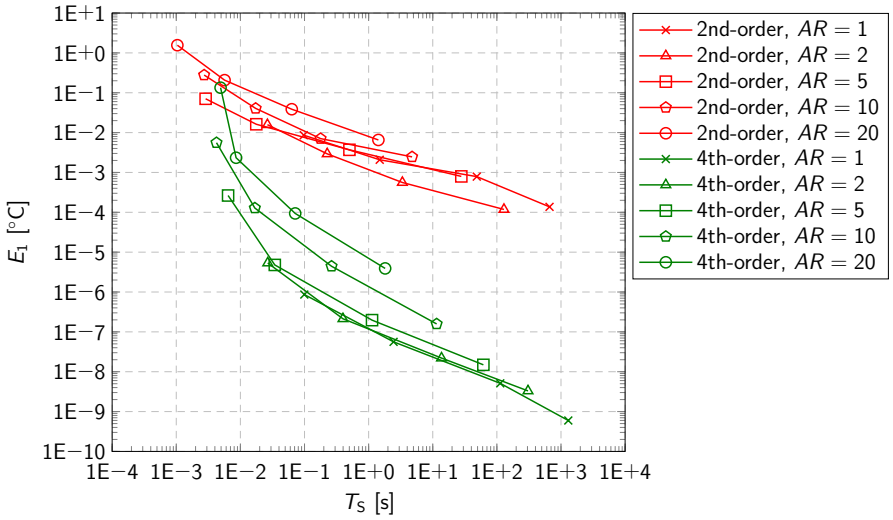


Figure 4.11: L^1-norm errors and execution time obtained in the annular section test case.

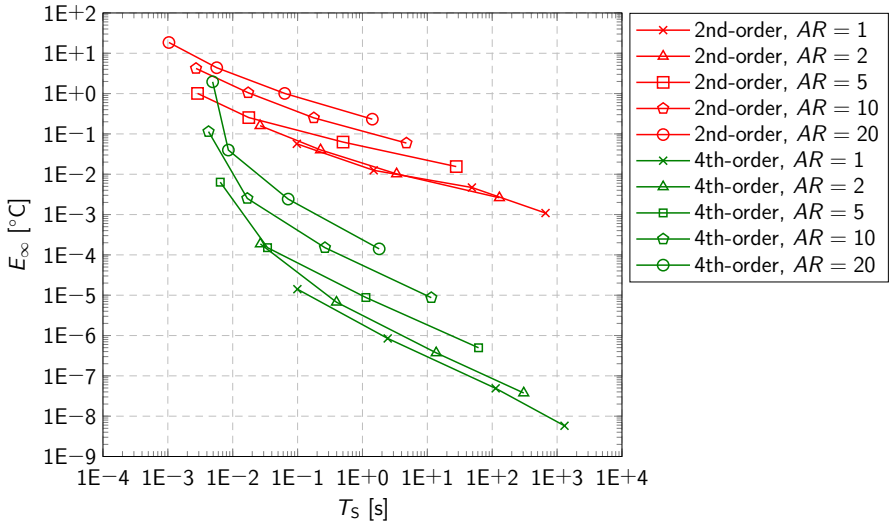


Figure 4.12: L^∞-norm errors and execution time obtained in the annular section test case.

with increasing aspect ratio of the mesh, both in the second- and fourth-orders of convergence. However, a more pronounced deterioration is observed in the latter. Nonetheless, meshes with higher aspect ratios can still further improve the efficiency of the fourth-order of convergence depending on the application requirements in what concerns accuracy.

For instance, with the finest mesh with $AR = 1$, the second- and fourth-orders of convergence provide an error in the L^1-norm of approximately $E_1 = 10^{-4}$ °C and $E_1 = 10^{-9}$ °C, respectively. Assuming that an error of $E_1 = 10^{-4}$ °C is required for a specific application, the fourth-order of convergence might be considered excessively accurate and with no practical benefits, when compared with the second-order of convergence, since latter requires half of the execution time. The standard approach consists in reducing the number of degrees of freedom to obtain the same accuracy, which usually provides a more efficient simulation given the better trade-off between accuracy and execution time of higher-orders of convergence. In that case, with the coarsest mesh and maintaining the aspect

ratio, the fourth-order of convergence provides an error of $E_1 = 10^{-6}$ °C, which is still significantly below the required accuracy for the application. Notice that, besides providing a significantly more accurate approximate solution, the associated execution time of $T_\mathrm{S} = 0.1$ s is already drastically lower than the execution time of approximately $T_\mathrm{S} = 663$ s for the second-order of convergence, meaning that improvement with a magnitude in the thousands is already obtained.

Further reduction in the number of cells maintaining the aspect ratio is not possible since a minimum number of 5 cells in the radial direction in each subdomain is required to compute the polynomial reconstructions. Therefore, the number of cells can only be reduced if the aspect ratio of the mesh increases. In that case, for the fourth-order of convergence, an error of approximately $E_1 = 10^{-4}$ °C is obtained with the coarsest mesh with $AR = 5$, with an associated execution time of roughly $T_\mathrm{S} = 0.01$ s. Therefore, an astonishing improvement with a magnitude in the tens of thousands is obtained, when compared with the second-order of convergence to achieve the same accuracy. Although in both cases of the fourth-order of convergence the execution time is imperceptible ($T_\mathrm{S} = 0.1$ s and $T_\mathrm{S} = 0.01$ s for the coarsest mesh with $AR = 1$ and $AR = 5$, respectively), there is a factor of at least 10, which can be relevant in intensive parametric investigations where thousands of simulations are carried out.

4.5 Case study

The thermoplastic sheet extrusion cooling stage is addressed solving numerically the model introduced in Section 4.2 with the discretization method introduced in Section 4.3. The dimensions of and the physical properties for the roll and the sheet are the following (see Figure 4.13). The roll has an external radius of $r^\mathrm{R} = 50$ mm, a thickness of $t^\mathrm{R} = 3$ mm, and is made of steel with thermal conductivity of $\kappa^\mathrm{R} = 30$ W/(m·K), density of $\rho^\mathrm{R} = 7700$ kg/m^3, and specific heat capacity of $c_\mathrm{p}^\mathrm{R} = 500$ J/(K·kg). The thermoplastic sheet has a thickness of $t^\mathrm{S} = 2$ mm and is made of polystyrene with thermal conductivity of $\kappa^\mathrm{S} = 0.2$ W/(m·K), density of $\rho^\mathrm{S} = 900$ kg/m^3, and specific heat capacity of $c_\mathrm{p}^\mathrm{S} = 1300$ J/(K·kg). Moreover, a distance of $d^\mathrm{I} = 40$ mm is considered from the inlet of the cooling stage to the roll. Moreover, the effective angle of contact between the roll and the sheet is approximately $\alpha \approx 0.8\pi$ rad, which results in a contact length of around 130 mm. After leaving the roll, a distance of approximately $d^\mathrm{O} = 34$ mm until the outlet of the cooling stage is considered, such that the variations of the temperature in the longitudinal direction of the sheet are negligible. The roll and the sheet have the same angular speed of $\omega^\mathrm{R} = \omega^\mathrm{S} = 0.2$ rad/s (the equivalent extrusion speed is 0.6 m/min) and, therefore, there is no discontinuity in the corresponding linear velocity provided with vector functions $\boldsymbol{u}^\mathrm{R}(\boldsymbol{x})$ and $\boldsymbol{u}^\mathrm{S}(\boldsymbol{x})$. To account for the effects of forced and natural heat convection, heat transfer coefficients of $h^\mathrm{R,W} = 10$ W/(K·m^2) between the roll and the water, $h^\mathrm{R,A} = 6$ W/(K·m^2) between the roll and the air, and $h^\mathrm{S,A} = 4$ W/(K·m^2) between the sheet and the air are considered. The temperature of the sheet on the inlet of the cooling stage is $T^\mathrm{I} = 180$ °C, the temperature of the water is $T_\infty^\mathrm{W} = 30$ °C, and the temperature of the air is $T_\infty^\mathrm{A} = 20$ °C.

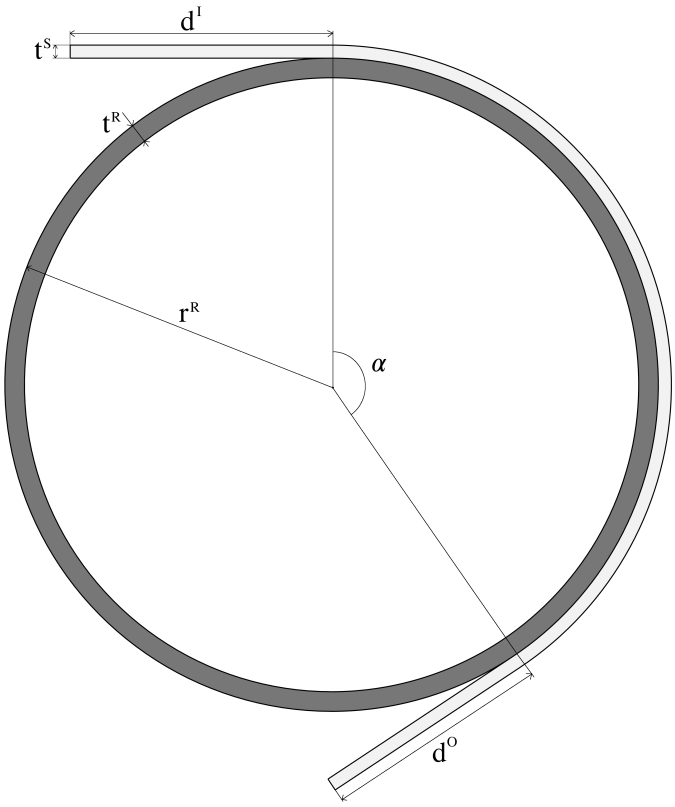


Figure 4.13: Geometry notation for the thermoplastic sheet extrusion cooling stage.

Successively finer structured quadrilateral meshes (with no curved elements) with aspect ratios of $AR = 1, 2, 5, 10$ are generated for each subdomain of the geometry (see Figures 4.14 to 4.17). The aspect ratio refers to the outer cells in the subdomain corresponding to the sheet, whereas the size of the remaining cells is adapted accordingly to obtain a structured mesh. For each aspect ratio, three successively finer meshes are generated with the coarsest denoted as M1, the medium as M2, and the finest as M3, having the number of cells presented in Table 4.4. Regardless of the aspect ratio, the coarsest, medium, and finest meshes have the same number of cells in the radial direction.

Table 4.4: Number of cells in the structured quadrilateral meshes for the thermoplastic sheet extrusion cooling stage.

Mesh	$AR = 1$	$AR = 2$	$AR = 5$	$AR = 10$
M1	8 122	4 248	1 859	960
M2	32 488	16 992	7 436	3 840
M3	129 952	67 968	29 744	15 360

Local mesh refinement is required at the initial contact point between the roll and sheet to provide sufficient resolution of the high temperature gradients arising in the vicinity and to avoid subsequent numerical instabilities and spurious solutions. Notice that Neumann boundary conditions as a function of the temperature are imposed immediately before the initial contact point, whereas

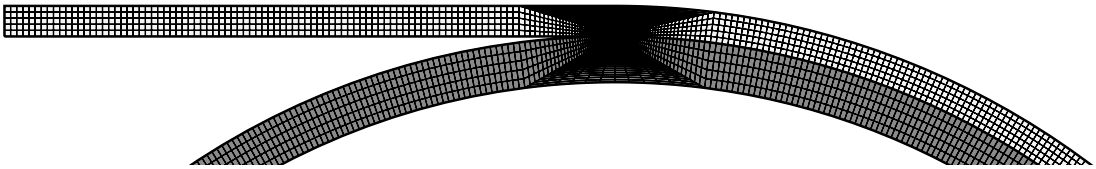


Figure 4.14: Detail of the coarsest structured quadrilateral mesh with $AR = 1$ for the thermoplastic sheet extrusion cooling stage.

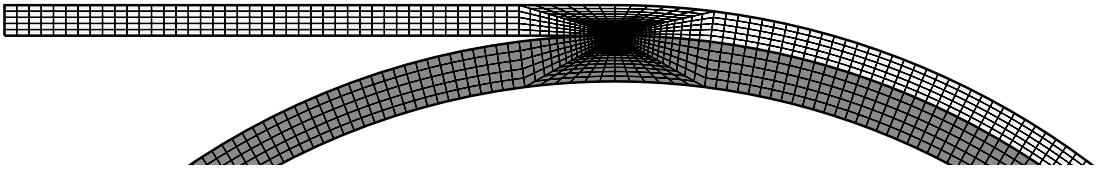


Figure 4.15: Detail of the coarsest structured quadrilateral mesh with $AR = 2$ for the thermoplastic sheet extrusion cooling stage.

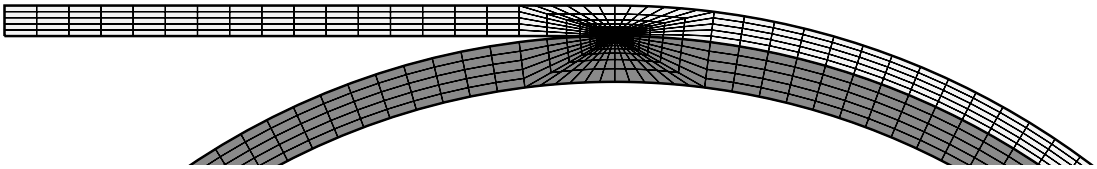


Figure 4.16: Detail of the coarsest structured quadrilateral mesh with $AR = 5$ for the thermoplastic sheet extrusion cooling stage.

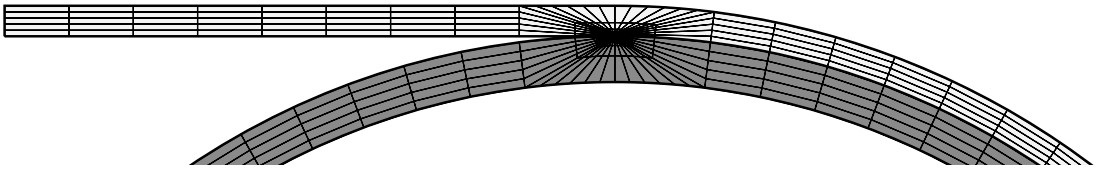


Figure 4.17: Detail of the coarsest structured quadrilateral mesh with $AR = 10$ for the thermoplastic sheet extrusion cooling stage.

the continuity interface conditions take place immediately after. Therefore, the initial contact point becomes a numerical singularity, although it does not represent an issue for the discretization method since boundary and interface conditions are not satisfied at the vertices of the mesh (or the corners of the geometry). However, such a situation does not allow a smooth temperature transition from the sheet to the roll and, therefore, the former has a substantially higher temperature than the latter. The same applies to the conductive heat flux since the natural heat convection implies different values on the boundaries of the roll and the sheet, although the conservation is imposed from the initial contact point forward. Notice that higher extrusion speeds result in higher temperature gradients and, therefore, the local mesh refinement should be applied accordingly.

The proposed numerical approach is verified assessing the accuracy and execution time provided with the second- and fourth-orders of convergence for the discretization method and meshes with the different aspect ratios. Since there is no exact solution for the thermoplastic sheet extrusion cooling stage, an approximate reference solution is computed considering the discretization method with a sixth-order of convergence and the finest mesh (M3) with $AR = 1$ (see Figure 4.18). From an industrial perspective, the thermoplastic sheet extrusion cooling stage has to ensure an average temperature below the solidification point and, simultaneously, a sufficiently uniform temperature distribution, to minimize the level of thermal residual stresses. Then, the optimization of the process seeks to satisfy these two conditions maximizing the cooling rate to increase productivity and achieve the desired physical properties of the sheet. In that regard, the verification of the proposed numerical approach consists in assessing the temperature distribution on boundary subset $\Gamma^{\mathrm{S,O}}$, which is assumed as the outlet of the cooling stage (see Figure 4.3). Since the outcome of the simulations consists of approximate cell mean-values, an interpolation method based on polynomial reconstructions is used to determine the temperature profile on boundary subset $\Gamma^{\mathrm{S,O}}$. The polynomial degree of the interpolation is set according to the convergence order of the discretization method to avoid accuracy deterioration, such that linear and cubic interpolations are used in the case of the second- and fourth-orders of convergence, respectively. The same interpolation procedure is used to provide a profile of the radial conductive heat flux on boundary subset $\Gamma^{\mathrm{S,O}}$ (in the normal direction the conductive heat flux given the prescribed homogeneous Neumann boundary condition). Since the conductive heat flux is a quantity derived from the temperature gradient, it also provides a measure of the level of the thermal residual stresses along with the thickness.

The obtained temperature and conductive heat flux profiles are provided in Figures 4.19 to 4.22 (r corresponds to the position along the width of the sheet starting from the inner corner). As observed, regardless of the aspect ratio of the mesh, both the temperature and the conductive heat flux converges asymptotically to the reference profiles as successively finer meshes are used. Moreover, the profiles obtained with the fourth-order of convergence are noticeably closer to the reference profiles when gather to those obtained with the second-order of convergence and the differences are difficult to perceive for lower aspect ratios. Therefore, these results strongly support that a discretization method with a higher-order of convergence does provide significantly more accurate approximate solutions for the same mesh. Increasing the aspect ratio of the mesh does indeed deteriorate the approximate solution accuracy, as expected from the results obtained in the code verification benchmark. It is particularly noticeable in the case of the second-order of convergence. Indeed, since the fourth-order of convergence provides significantly more accurate approximate solutions, beyond the requirements of the application, the visual perception of the accuracy deterioration when the aspect ratio of the mesh increases is less evident. Therefore, contrary to the second-order of convergence, higher-orders of convergence have the capability of reducing the computational effort using meshes with higher aspect ratios without perceived accuracy deterioration.

The average and standard deviation associated with the previous temperature and conductive

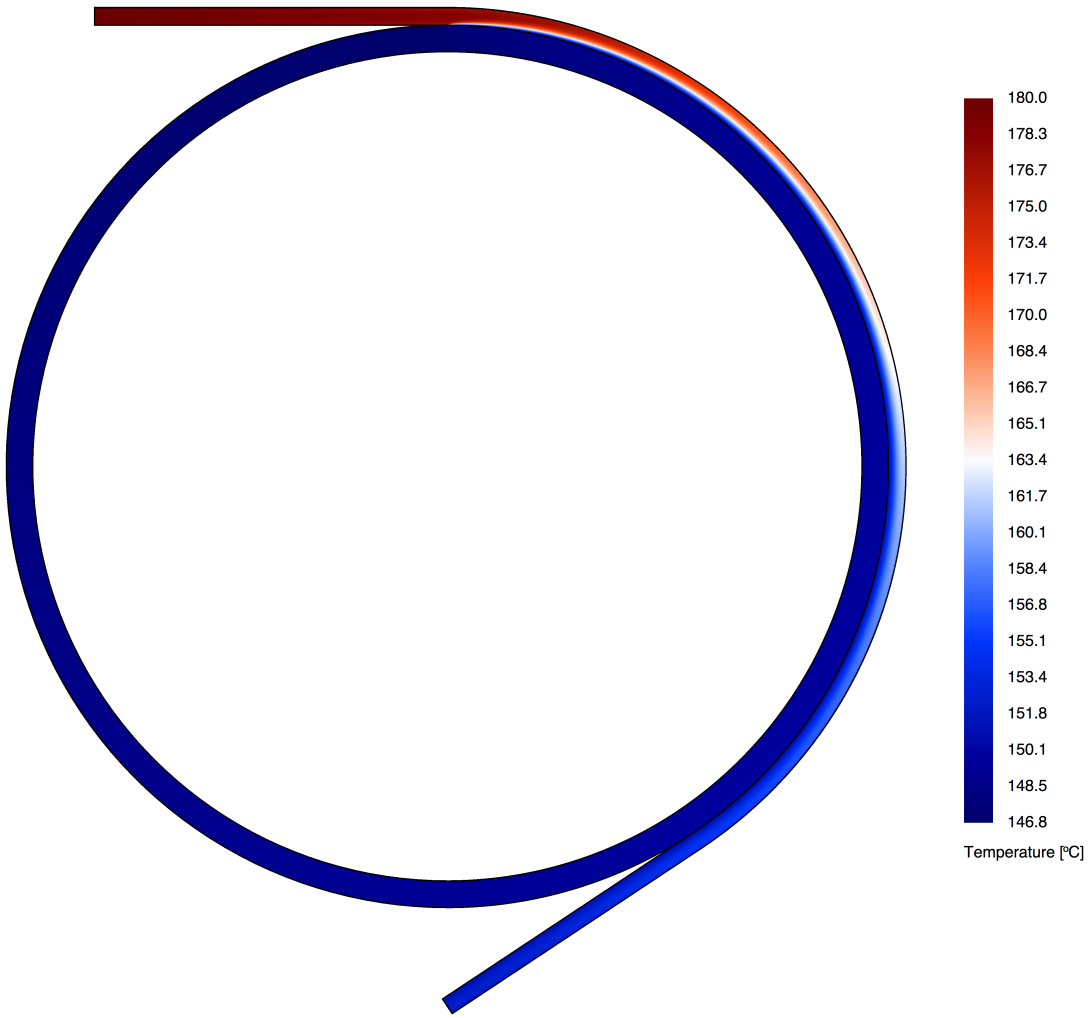


Figure 4.18: Reference solution for the thermoplastic sheet extrusion cooling stage.

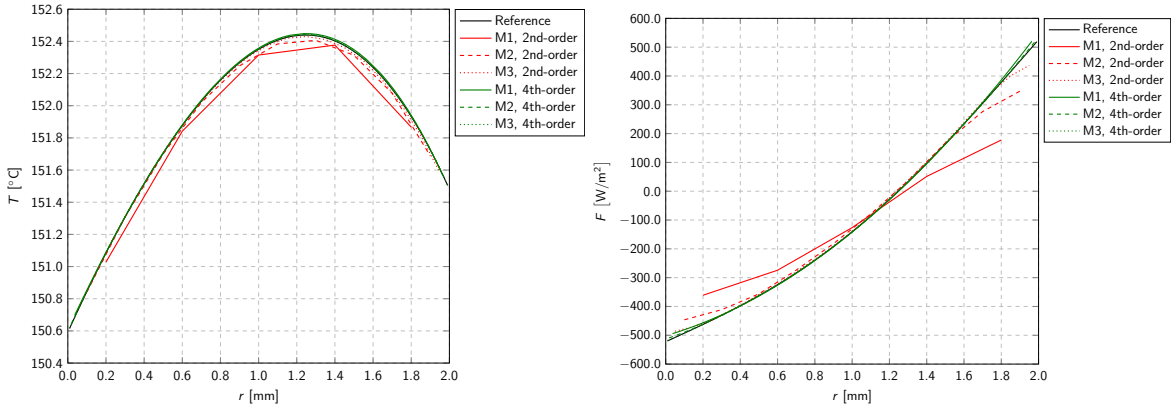
heat flux profiles are determined, and the corresponding errors are provided, having the values obtained from the reference profiles, as follows. In exact terms, the average and standard deviation of the temperature on boundary subset $\Gamma^{\mathrm{S,O}}$, denoted as $\overline{T}^{\mathrm{E}}$ and σ_T^{E}, respectively, are given as

$$\overline{T}^{\mathrm{E}} = \frac{1}{\left|\Gamma^{\mathrm{S,O}}\right|} \int_{\Gamma^{\mathrm{S,O}}} T^{\mathrm{S}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{4.33}$$

$$\sigma_T^{\mathrm{E}} = \left(\frac{1}{\left|\Gamma^{\mathrm{S,O}}\right|} \int_{\Gamma^{\mathrm{S,O}}} \left(T^{\mathrm{S}}(\boldsymbol{x}) - \overline{T}^{\mathrm{E}} \right)^2 \mathrm{d}\boldsymbol{x} \right)^{1/2}, \tag{4.34}$$

where $\left|\Gamma^{\mathrm{S,O}}\right|$ stands for the length of the boundary subset, the same as the sheet thickness. Similarly, the average and standard deviation of the conductive heat flux on boundary subset $\Gamma^{\mathrm{S,O}}$, denoted as $\overline{F}^{\mathrm{E}}$ and σ_F^{E}, respectively, are given as

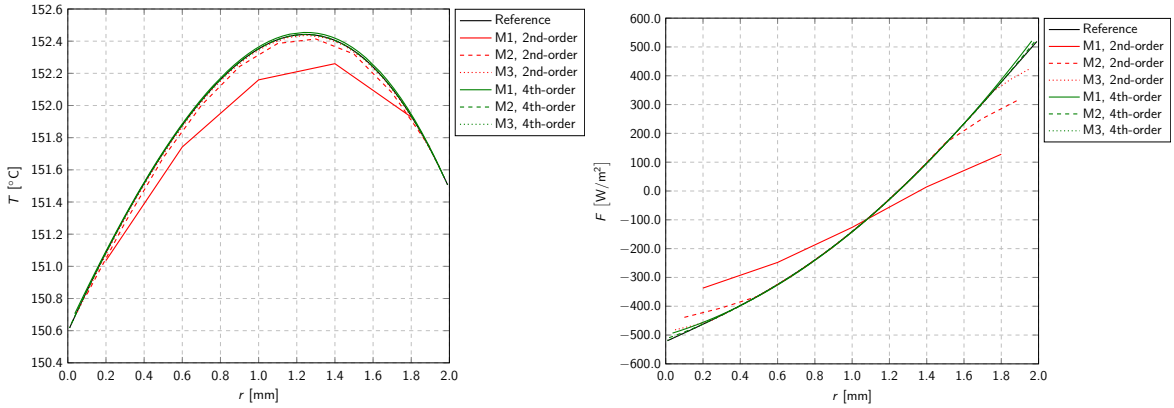
$$\overline{F}^{\mathrm{E}} = \frac{1}{\left|\Gamma^{\mathrm{S,O}}\right|} \int_{\Gamma^{\mathrm{S,O}}} \left(-\kappa^{\mathrm{S}} \nabla T^{\mathrm{S}}(\boldsymbol{x}) \cdot \boldsymbol{t}^{\mathrm{S,O}} \right) \mathrm{d}\boldsymbol{x}, \tag{4.35}$$



(a) Temperature profile.

(b) Conductive heat flux profile.

Figure 4.19: Temperature and conductive heat flux profiles with meshes with $AR = 1$ for the thermoplastic sheet extrusion cooling stage.



(a) Temperature profile.

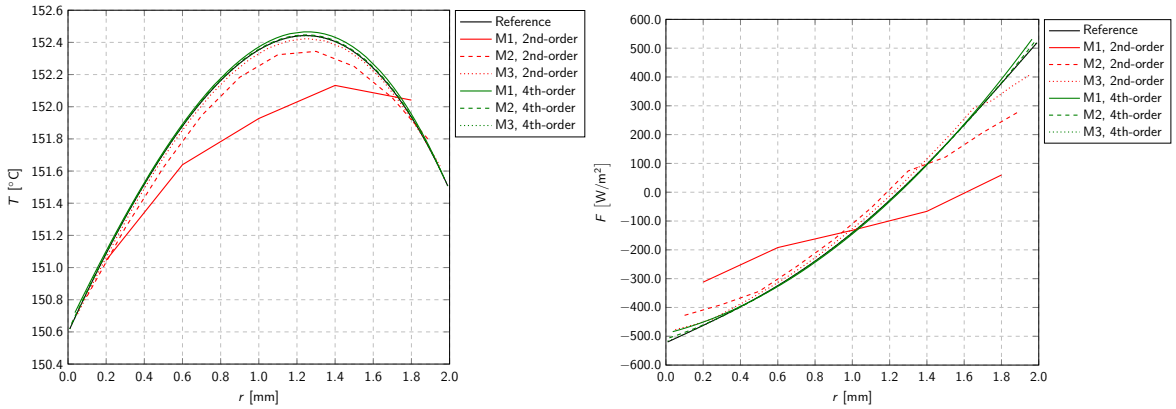
(b) Conductive heat flux profile.

Figure 4.20: Temperature and conductive heat flux profiles with meshes with $AR = 2$ for the thermoplastic sheet extrusion cooling stage.

$$\sigma_F^{\mathrm{E}} = \left(\frac{1}{\left| \Gamma^{\mathrm{S,O}} \right|} \int_{\Gamma^{\mathrm{S,O}}} \left(-\kappa^{\mathrm{S}} \nabla T^{\mathrm{S}}(\boldsymbol{x}) \cdot \boldsymbol{t}^{\mathrm{S,O}} - \overline{F}^{\mathrm{E}} \right)^2 \mathrm{d}\boldsymbol{x} \right)^{1/2}, \tag{4.36}$$

where $\boldsymbol{t}^{\mathrm{S,O}}$ stands for the tangential vector on boundary subset $\Gamma^{\mathrm{S,O}}$.

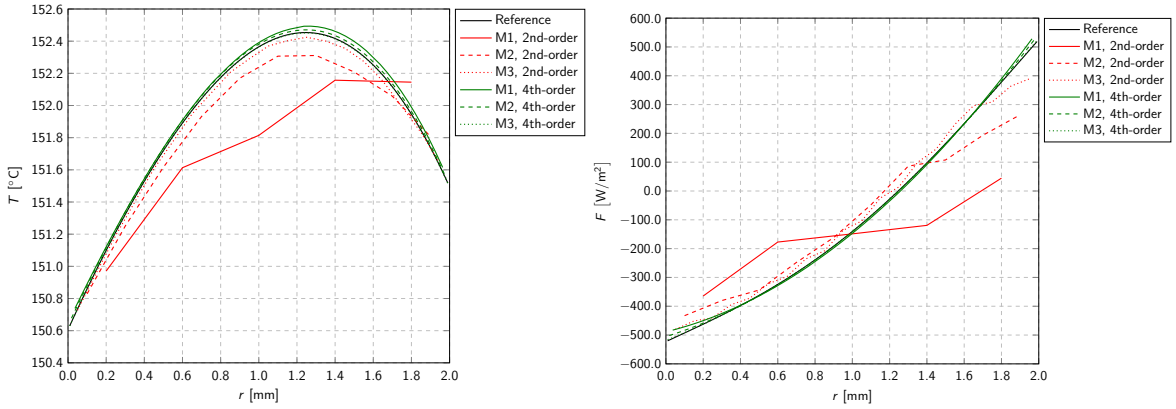
Consider edges e_{iF} on boundary subset $\Gamma^{\mathrm{S,O}}$ with length $|e_{iF}|$, tangent vector $\boldsymbol{t}_{iF}$, and quadrature points $\boldsymbol{q}_{iF,r}$, $r = 1, \ldots, R$, for a Gaussian quadrature rule with weights ξ_r, where R is the number of quadrature points required to obtain the same convergence order as the discretization method. For each edge, a polynomial reconstruction, denoted as $\widetilde{T}_{iF}(\boldsymbol{x})$, is computed and corresponds to the best approximation in the least-squares sense of the cell mean-values in the vicinity provided in the approximate solution. Given that, the approximate average and standard deviation of the



(a) Temperature profile.

(b) Conductive heat flux profile.

Figure 4.21: Temperature and conductive heat flux profiles with meshes with $AR = 5$ for the thermoplastic sheet extrusion cooling stage.



(a) Temperature profile.

(b) Conductive heat flux profile.

Figure 4.22: Temperature and conductive heat flux profiles with meshes with $AR = 10$ for the thermoplastic sheet extrusion cooling stage.

temperature on boundary subset $\Gamma^{\mathrm{S,O}}$, denoted as $\overline{T}$ and σ_T, respectively, are given as

$$\overline{T} = \frac{1}{\left|\Gamma^{\mathrm{S,O}}\right|} \sum_{e_{iF} \in \Gamma^{\mathrm{S,O}}} |e_{iF}| \left[\sum_{r=1}^{R} \xi_r \widetilde{T}_{iF}\left(\boldsymbol{q}_{iF,r}\right) \right], \tag{4.37}$$

$$\sigma_T = \left(\frac{1}{\left|\Gamma^{\mathrm{S,O}}\right|} \sum_{e_{iF} \in \Gamma^{\mathrm{S,O}}} |e_{iF}| \left[\sum_{r=1}^{R} \xi_r \left(\widetilde{T}_{iF}\left(\boldsymbol{q}_{iF,r}\right) - \overline{T} \right)^2 \right] \right)^{1/2}. \tag{4.38}$$

Similarly, the approximate average and standard deviation of the conductive heat flux on boundary subset $\Gamma^{\mathrm{S,O}}$, denoted as $\overline{F}$ and σ_F, respectively, are given as

$$\overline{F} = \frac{1}{\left|\Gamma^{\mathrm{S,O}}\right|} \sum_{e_{iF} \in \Gamma^{\mathrm{S,O}}} |e_{iF}| \left[\sum_{r=1}^{R} \xi_r \left(-\kappa^{\mathrm{S}} \nabla \widetilde{T}_{iF}\left(\boldsymbol{q}_{iF,r}\right) \cdot \boldsymbol{t}_{iF} \right) \right], \tag{4.39}$$

$$\sigma_F = \left(\frac{1}{\left|\Gamma^{\mathrm{S,O}}\right|} \sum_{e_{iF} \in \Gamma^{\mathrm{S,O}}} |e_{iF}| \left[\sum_{r=1}^{R} \xi_r \left(-\kappa^{\mathrm{S}} \nabla \widetilde{T}_{iF}(\boldsymbol{q}_{iF,r}) \cdot \boldsymbol{t}_{iF} - \overline{F} \right)^2 \right] \right)^{1/2}. \tag{4.40}$$

In particular, for the reference solution, the reference average and standard deviation of the temperature are denoted as $\overline{T}^*$ and σ_T^*, respectively, whereas for the conductive heat flux are denoted as $\overline{F}^*$ and σ_F^*, respectively. Then, for each simulation, the errors of the average and standard deviation of the temperature, denoted as $E_{\overline{T}}$ and E_{σ_T}, respectively, and of the conductive heat flux, denoted as $E_{\overline{F}}$ and E_{σ_F}, respectively, are given as

$$E_{\overline{T}} = \overline{T} - \overline{T}^*, \tag{4.41}$$

$$E_{\sigma_T} = \sigma_T - \sigma_T^*, \tag{4.42}$$

$$E_{\overline{F}} = \overline{F} - \overline{F}^*, \tag{4.43}$$

$$E_{\sigma_F} = \sigma_F - \sigma_F^*, \tag{4.44}$$

(4.45)

Then, the relative errors with respect to the reference average and standard deviation are also provided, in percentage, denoted as $R_{\overline{T}}$, R_{σ_T}, $R_{\overline{F}}$, and R_{σ_F}, respectively, and given as

$$R_{\overline{T}} = \frac{E_{\overline{T}}}{\overline{T}^*} \times 100, \tag{4.46}$$

$$R_{\sigma_T} = \frac{E_{\sigma_T}}{\sigma_T^*} \times 100, \tag{4.47}$$

$$R_{\overline{F}} = \frac{E_{\overline{F}}}{\overline{F}^*} \times 100, \tag{4.48}$$

$$R_{\sigma_F} = \frac{E_{\sigma_F}}{\sigma_F^*} \times 100. \tag{4.49}$$

To determine the errors, the reference values obtained for the average and standard deviation of the temperature and conductive heat flux are $\overline{T}^* = 151.911727$ °C, $\sigma_T^* = 0.503300$ °C, $\overline{F}^* = -88.893056$ W/m^2, and $\sigma_F^* = 366.387595$ W/m^2. Then, the errors, relative errors, and execution time are reported in Table 4.5 for meshes with $AR = 1$. As observed, the fourth-order of convergence provides significantly more accurate results in terms of average and standard deviation of both the temperature and the conductive heat flux than the second-order of convergence. In general, the errors convergence under mesh refinement, which is in accordance with the behavior observed for the profiles in Figures 4.19 to 4.22. Both the second- and fourth-orders of convergence provide relative errors of the average and standard deviation of the temperature below 10%, even with the coarsest mesh, although the latter is significantly more accurate. However, the second-order of convergence provides relative errors of approximately 20% and 14% for the average and standard

deviation, respectively, of the conductive heat flux with the coarsest mesh. On the other side, the fourth-order of convergence provides corresponding relative errors of approximately 3% and 6%, which are already below the limit of 10%. The second-order of convergence achieves such accuracy with the medium mesh, taking approximately 111 s for the simulation, whereas the fourth-order of convergence takes only 10 s with the coarsest mesh. Therefore, for the same accuracy level, the fourth-order of convergence allows a significant computational gain with a ratio of improvement of around 11. Although the second-order of convergence takes only a few minutes, which is feasible for a single simulation, such computational gain would have a significant impact in intensive parametric investigations where thousands of simulations are required. Moreover, notice that the computational gain is more pronounced as more accurate results are required for the thermoplastic sheet extrusion cooling stage.

The errors, relative errors, and execution time are reported in Tables 4.6 to 4.8 for meshes with $AR = 2, 5, 10$, respectively. In general, increasing the aspect ratio of the mesh does deteriorate the accuracy of the average and standard deviation of both the temperature and the conductive heat flux, which follows the conclusions drawn in the code verification benchmark. However, as in the case of meshes with $AR = 1$, the fourth-order of convergence is capable of still providing relative errors below 10%, contrarily to the second-order of convergence. For instance, to provide a relative error of both the average and the standard deviation of the conductive heat flux below 10%, the second-order of convergence requires the finest mesh with $AR = 10$. On the other side, the coarsest mesh is sufficient for the fourth-order of convergence, taking approximately 0.05 s in comparison with around 14 s in the case of the second-order of convergence. Once again, a significant computational gain is achieved with a noticeable ratio of approximately 280. Moreover, even if there is some level of accuracy deterioration, increasing the aspect ratio of the mesh for the fourth-order of convergence does reduce the computational cost when the provided error still falls below the accuracy requirements for the application. Notice that these results follow the conclusions drawn in the code verification benchmark. Therefore, generating meshes with a higher aspect ratio for the fourth-order of convergence is a useful strategy to improve further the computational efficiency of the proposed numerical approach for the thermoplastic sheet extrusion cooling stage.

Although the accuracy of the average and standard deviation of the temperature does not seem critical, regardless of the convergence order or the aspect ratio of the mesh, the fourth-order of convergence is still capable of providing significantly more accurate results. Therefore, it turns out more reliable and safe to carry out the simulations for the thermoplastic sheet extrusion cooling stage with a higher-order of convergence. However, higher extrusion speeds and non-linear conjugate heat transfer models with temperature-dependent physical properties, such as the thermal conductivity, might deteriorate the approximate solution accuracy and, in that scenario, the benefits of using higher-orders of convergence to predict the average and standard deviation of the temperature are more evident. Nevertheless, these scenarios require further developments and investigations.

Table 4.5: Errors, relative errors, and execution time obtained with meshes with $AR = 1$ for the thermoplastic sheet extrusion cooling stage.

Mesh	$E_{\overline{T}}$ [°C]	$R_{\overline{T}}$ [%]	E_{σ_T} [°C]	R_{σ_T} [%]	$E_{\overline{F}}$ [W/m^2]	$R_{\overline{F}}$ [%]	E_{σ_F} [W/m^2]	R_{σ_F} [%]	T_S [s]
Second-order of convergence									
M1	−2.60E−02	0.017	−3.56E−03	0.708	−1.76E+01	19.778	−4.967E+01	13.555	5.68
M2	−1.29E−02	0.009	−8.04E−03	1.598	−3.12E+00	3.506	−1.744E+01	4.760	111.13
M3	−5.47E−03	0.004	−3.78E−03	0.751	6.24E−01	0.702	−8.546E+00	2.332	3 105.59
Fourth-order of convergence									
M1	7.13E−03	0.005	1.24E−03	0.245	2.42E+00	2.724	2.319E+01	6.331	10.41
M2	2.46E−03	0.002	−2.59E−05	0.005	4.51E−01	0.507	9.639E+00	2.631	159.46
M3	7.38E−04	<0.001	2.10E−05	0.004	3.81E−02	0.043	−6.911E−03	0.002	4 061.44

Table 4.6: Errors, relative errors, and execution time obtained with meshes with $AR = 2$ for the thermoplastic sheet extrusion cooling stage.

Mesh	$E_{\overline{T}}$ [°C]	$R_{\overline{T}}$ [%]	E_{σ_T} [°C]	R_{σ_T} [%]	$E_{\overline{F}}$ [W/m^2]	$R_{\overline{F}}$ [%]	E_{σ_F} [W/m^2]	R_{σ_F} [%]	T_S [s]
Second-order of convergence									
M1	−8.66E−02	0.057	−5.38E−02	10.684	−2.49E+01	28.051	−6.311E+01	17.225	0.65
M2	−1.66E−02	0.011	−6.61E−04	0.131	−1.15E+01	12.957	−1.421E+01	3.879	15.70
M3	−2.74E−03	0.002	1.27E−03	0.252	−3.65E+00	4.108	−6.553E+00	1.788	410.86
Fourth-order of convergence									
M1	1.26E−02	0.008	1.05E−03	0.209	2.64E+00	2.971	2.289E+01	6.247	1.11
M2	2.33E−03	0.002	1.13E−04	0.023	4.38E−01	0.492	9.820E+00	2.680	21.44
M3	1.63E−03	0.001	−3.11E−05	0.006	3.46E−02	0.039	−4.185E−03	0.001	712.68

4.6 Conclusions

A numerical approach is proposed and verified to investigate the thermodynamics in the thermoplastic sheet extrusion cooling stage. The approach consists in solving a steady-state two-dimensional conjugate heat transfer problem with the appropriate boundary and interface conditions between a polymeric sheet and a metallic roll. The problem is discretized with a finite volume method capable of providing a very high-order of convergence, which improves the accuracy of the approximate solutions and reduce the computational cost of the simulations. The degree of the polynomial reconstructions in the discretization method determines the convergence order, such that for a degree d an optimal $(d+1)$th-order of convergence is expected.

A code verification benchmark based on manufactured solutions is provided to assess the accuracy, convergence orders, and execution time of the discretization method according to the aspect ratio of the mesh. The results obtained prove that for polynomial degrees $d = 1, 3$ the optimal

Table 4.7: Errors, relative errors, and execution time obtained with meshes with $AR = 5$ for the thermoplastic sheet extrusion cooling stage.

Mesh	$E_{\overline{T}}$ [°C]	$R_{\overline{T}}$ [%]	E_{σ_T} [°C]	R_{σ_T} [%]	$E_{\overline{F}}$ [W/m^2]	$R_{\overline{F}}$ [%]	E_{σ_F} [W/m^2]	R_{σ_F} [%]	T_S [s]
Second-order of convergence									
M1	−1.55E−01	0.102	−9.74E−02	19.360	−3.97E+01	44.622	−7.029E+01	19.184	0.12
M2	−5.44E−02	0.036	−2.12E−02	4.210	−6.42E+00	7.228	−3.882E+01	10.595	1.76
M3	−1.30E−02	0.009	−8.58E−04	0.171	4.79E+00	5.394	−2.046E+01	5.583	55.54
Fourth-order of convergence									
M1	2.17E−02	0.014	1.46E−03	0.289	3.32E+00	3.738	2.256E+01	6.157	0.10
M2	7.50E−03	0.005	−1.11E−04	0.022	9.43E−01	1.061	9.577E+00	2.614	2.91
M3	2.28E−03	0.002	−1.98E−04	0.039	5.43E−02	0.061	1.440E−02	0.004	93.80

Table 4.8: Errors, relative errors, and execution time obtained with meshes with $AR = 10$ for the thermoplastic sheet extrusion cooling stage.

Mesh	$E_{\overline{T}}$ [°C]	$R_{\overline{T}}$ [%]	E_{σ_T} [°C]	R_{σ_T} [%]	$E_{\overline{F}}$ [W/m^2]	$R_{\overline{F}}$ [%]	E_{σ_F} [W/m^2]	R_{σ_F} [%]	T_S [s]
Second-order of convergence									
M1	−1.72E−01	0.113	−5.19E−02	10.303	−6.41E+01	72.134	−2.547E+01	6.951	0.04
M2	−6.29E−02	0.041	−3.26E−02	6.483	−7.34E+00	8.257	−4.301E+01	11.738	0.44
M3	−1.03E−02	0.007	−7.55E−03	1.500	8.04E+00	9.045	−2.634E+01	7.190	14.21
Fourth-order of convergence									
M1	4.34E−02	0.029	4.40E−03	0.873	2.58E+00	2.898	2.289E+01	6.247	0.05
M2	3.12E−02	0.021	−3.53E−04	0.070	1.53E+00	1.719	9.326E+00	2.545	0.83
M3	1.24E−02	0.008	−5.35E−04	0.106	1.55E−01	0.174	−3.497E−02	0.010	19.54

second- and fourth-orders of convergence are effectively achieved. Moreover, the latter provides significantly more accurate approximate solutions with the same mesh. Furthermore, the fourth-order of convergence provides approximate solutions with the same accuracy as the second-order of convergence but with significantly coarser meshes, which turns out to be computationally more efficient. Therefore, the better trade-off between accuracy and computational cost of higher-orders of convergence allows improving the efficiency of the proposed numerical approach. On the other side, accuracy deterioration is observed when increasing the aspect ratio of the mesh in both convergence orders since the reconstruction matrices have higher condition numbers. However, when the accuracy provided with meshes with unitary aspect ratio falls below the application requirements, increasing the aspect ratio is a valid strategy to improve the efficiency with the fourth-order of convergence further.

Several simulations for the thermoplastic sheet extrusion cooling stage are carried out to assess the capabilities of the proposed numerical approach according to the convergence order of the

discretization method and the aspect ratio of the mesh. The profile, average, and standard deviation of both the temperature and the conductive heat flux on the outlet of the cooling stage are used for verification purposes. Moreover, the errors associated with the average and standard deviation are determined, and the execution time is reported for each simulation. The results support that, for the same mesh, the fourth-order of convergence provides a significantly more accurate average and standard deviation of both the temperature and the conductive heat flux than the second-order of convergence. Moreover, the fourth-order of convergence provides the same accuracy as the second-order of convergence with coarser meshes, which turns out to be computationally more efficient. Additionally, meshes with higher aspect ratios allow to improve further the efficiency of the proposed numerical approach with the fourth-order of convergence and provide the same accuracy as the second-order of convergence. Using a discretization method with higher-orders of convergence and meshes with higher aspect ratios, the proposed numerical approach is an efficient computational tool to investigate the heat exchanges in the thermoplastic sheet extrusion cooling stage.

Further developments might comprise polymeric materials having temperature-dependent physical properties considering non-linear conjugate heat transfer models and parametric investigations on the thermoplastic sheet extrusion cooling stage, aiming at the optimization of the process. In any case, an increased computational effort is required, for which the proposed numerical approach will be a valuable tool.

References

[1] D.V. Rosato, Extruding plastics: a practical processing handbook, Springer Science & Business Media (1998).

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

[3] C. Rauwendaal, R. Gonzalez-Nunez, D. Rodrigue, Polymer Processing: Extrusion, in Encyclopedia of Polymer Science and Technology, John Wiley & Sons (2017) 1–67.

[4] E.M. Abdel-Bary, (ed.), Handbook of plastic films, iSmithers Rapra Publishing (2003).

[5] J. Vlachopoulos, N. Polychronopoulos, S. Tanifuji, J. Peter Müller, Chapter 4: Flat film and sheet dies, in Design of extrusion forming tools, O.S. Carneiro, J.M. Nóbrega (eds.), Smithers Rapra, London (2012) 113–140.

[6] N.J. Macauley, E.M.A. Harkin-Jones, W.R. Murphy, The influence of extrusion parameters on the mechanical properties of polypropylene sheet, Polym. Engrg. Sci. 38 (4) (1998) 662–673.

[7] C. Sollogoub, E. Felder, Y. Demay, J.F. Agassant, P. Deparis, N. Mikler, Thermomechanical analysis and modeling of the extrusion coating process, Polym. Engrg. Sci. 48 (8) (2008) 1634–1648.

[8] O.S. Carneiro, J.M. Nóbrega, Chapter 1: Main issues in the design of extrusion tools, in Design of extrusion forming tools, O. Carneiro, J.M. Nóbrega (eds.), Smithers Rapra, London (2012) 1–36.

[9] M.L. Whitefield, Sheet extrusion for today's products, J. Past. Film Sheet. 6 (1990) 10–16.

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

[11] 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.

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

[13] R. Costa, S. Clain, R. Loubère, G.J. Machado, Very high-order accurate polygonal mesh finite volume scheme for conjugate heat transfer problems with curved interfaces and imperfect contacts, Comput. Meth. Appl. Mech. Engrg. 357 (2019) 112560.

[14] A. Jalali, C.F. Ollivier-Gooch, Higher-order finite volume solution reconstruction on highly anisotropic meshes, 21st AIAA Computational Fluid Dynamics Conference (2013).

CHAPTER 5 Efficient very high-order accurate polyhedral mesh finite volume scheme for 3D conjugate heat transfer problems in curved domains

Abstract: The conjugate heat transfer problem is found in many engineering applications consisting of non-isothermal solids and fluids in contact with different physical properties. The discretization method of these problems must guarantee that the interface conditions are properly satisfied, which becomes challenging to obtain a very high-order of convergence. The three-dimensional case with arbitrary curved domains is particularly challenging to treat as the standard approaches to preserve the optimal converge order become cumbersome with the use of curved meshes and non-linear transformations. Moreover, the implementation of discretization methods with very high-order of convergence is critical for the three-dimensional case as the computational cost of conventional algorithms often grows exponentially with the problem dimension. A very high-order accurate finite volume scheme in general polyhedral meshes is proposed to solve three-dimensional conjugate heat transfer problems in arbitrary curved domains. The use of polyhedral mesh elements overcomes the shortcomings of the standard curved mesh approaches, preserves the accuracy and the convergence order, is more simple to implement, and is computationally efficient. Moreover, the implementation of the proposed method is addressed and optimized to provide the same approximate solution at a reduced computational cost. A comprehensive numerical benchmark addressing several situations of the conjugate heat transfer problem is provided and verifies that the optimal convergence order is effectively achieved. A computational benchmark proves that substantial performance improvements are obtained from the proposed optimization, which represents a significant advance towards less demanding and more efficient numerical simulations.

Keywords: Conjugate heat transfer problems, Arbitrary curved three-dimensional domains, Polyhedral unstructured meshes, Polynomial reconstruction method, Very high-order accurate finite volume scheme, Implicit and explicit formulation

This chapter was adapted from R. Costa, J.M. Nóbrega, S. Clain, and G.J. Machado, Efficient very high-order accurate polyhedral mesh finite volume scheme for 3D conjugate heat transfer problems in curved domains, in preparation for J. Comput. Phys. (2020)

5.1 Introduction

The conjugate heat transfer is found in many engineering problems. It corresponds to the heat transfer in solid-solid, solid-fluid, and fluid-fluid interfaces between non-isothermal materials with different physical properties. The conjugate model consists in solving the convective and conductive transport of thermal energy in each material domain subjected to interface conditions. For perfect thermal contacts, these interface conditions impose the continuity of the temperature and the conservation of the heat flux, whereas other interface conditions are used for imperfect thermal contacts. Comprehensive literature is found concerning numerical methods to solve conjugate heat transfer problems and, in general, discretization methods with first- and second-orders of convergence are easily accessible and practical to implement. However, obtaining very high-order of convergence (here defined as more than the second-order of convergence) becomes a challenging task since the interface conditions have to be properly satisfied within the same error magnitude usually requires more elaborated methods.

The three-dimensional case with arbitrary curved domains is particularly challenging to treat since the conventional discretization methods do not handle the geometrical mismatch properly between the mesh and the boundaries and interfaces, providing a second-order of convergence at most. In that regard, several approaches arise to preserve the optimal convergence order. The most common is the isoparametric elements in the finite element method that consists in using curved mesh elements to follow the curved boundaries and interfaces, but also require non-linear transformations for the finite element basis and quadrature rules for the integration in the curved mesh elements [1, 2]. Unfortunately, generating curved meshes for arbitrary domains require cumbersome and computationally intensive algorithms [3–6], the non-linear transformations are complex to perform, and the quadrature rules are difficult to obtain [7–11], which becomes even more challenging in the three-dimensional case.

The development of approaches that only rely on polygonal or polyhedral mesh elements is a recent topic of increasing interest to avoid the drawbacks of methods such as the isoparametric elements. Many techniques have been developed, namely within the immersed boundary method, and a comprehensive literature review on the topic is found in Fernández-Fidalgo et al., 2020 [12]. In the framework of the finite volume discretization, the reconstruction for off-site data (ROD) method is a recent approach proposed in Costa et al., 2018 [13, 14], and is capable of preserving the optimal convergence order using polygonal meshes in arbitrary curved domains. The method consists in computing polynomial reconstructions from the least-squares method constrained with specific linear equations to fulfil the prescribed boundary conditions at collocation points on the associated curved boundaries.

The method was successfully applied to solve conjugate heat transfer problems in two-dimensional domains with perfect and imperfect thermal contacts [15] but the three-dimensional case has not been addressed yet. Moreover, the literature concerning the three-dimensional

conjugate heat transfer problems case is very scarce, where no method is known to provide a very high-order convergence regardless of the geometry of the domain. Indeed, the three-dimensional case represents an increased challenge in what concerns the implementation of the methods and the need for powerful computational resources for the simulations. Nevertheless, realistic problems in engineering not always can be reduced to two-dimensions, making it imperative to extend and verify such approaches in three-dimensional geometries carefully for practical use. Moreover, the implementation of the methods is also a critical point in the three-dimensional case given the exponential growth of the computational cost with the increasing dimension of the problem. Therefore, careful optimization of the algorithm is of crucial importance to obtain substantial computational gains in terms of execution time and memory usage, also demanding less powerful computational resources and lower energy expenditure.

The novelties of the present work are split into a numerical part and computational part. The numerical part consists in the extension to the three-dimensional case of the very high-order accurate finite volume method to solve the steady-state conjugate heat transfer problem in arbitrary curved domains. Moreover, the Robin boundary condition is used as a general approach to prescribing the fundamental Dirichlet and Neumann boundary conditions assigning the appropriate coefficients. The computational part consists in addressing the implementation of the proposed method in terms of computational efficiency and provide performance improvements that reduce the execution time and the memory usage of the simulations. The optimization consists in deriving equations for the several steps of the implicit formulation of the proposed method that provide the same result at a reduced computational cost. Moreover, an explicit formulation of the proposed method providing a coefficients matrix and right-hand side vector is obtained from the optimization.

The remaining sections of the chapter are organized as follows. Section 2 presents the conjugate problem model and the equivalent partitioned problem model with the appropriate thermal coupling. Section 3 is dedicated to the finite volume discretization of the partitioned problem model and the numerical treatment of the thermal coupling. Section 4 provides an optimization of the implementation of the proposed method. Section 5 and Section 6 are dedicated to a comprehensive numerical and computational benchmark, respectively. The chapter is completed in Section 7 with the conclusions and some perspectives for future work.

5.2 Mathematical formulation

The steady-state conjugate heat transfer problem is addressed in three dimensions and formulated with the Cartesian coordinate system considering $\boldsymbol{x} := (x, y, z)$. For the sake of simplicity and without loss of generality, a partition with only two subdomains is considered and treated. Still, the problem is easily extended for the case of multiple subdomains.

5.2.1 Conjugate problem model

Let Ω denote an open bounded arbitrary curved domain in $\mathbb{R}^3$ with boundary Γ^{R}, partitioned into two non-overlapping subdomains, Ω^S, $S \in \{\mathrm{A},\mathrm{B}\}$, sharing a common interface, denoted as Γ^{C}. The boundary and the interface are regular and closed surfaces and admit a regular local parameterization. Boundary Γ^{R} is partitioned into two non-overlapping subsets (where one of the two can be the empty set as in the case of two concentric subdomains), denoting as $\Gamma^{\mathrm{R},S}$ the boundary of subdomain Ω^S, such that $\Gamma^{\mathrm{R}} = \Gamma^{\mathrm{R,A}} \cup \Gamma^{\mathrm{R,B}}$. For each subdomain Ω^S, let Γ^S denote the respective boundary, consisting of boundaries and interface subsets of domain Ω, namely $\Gamma^S = \Gamma^{\mathrm{R},S} \cup \Gamma^{\mathrm{C}}$. Moreover, for any point $\boldsymbol{x}$ on boundary Γ^S, vector $\boldsymbol{n}^S := \boldsymbol{n}^S(\boldsymbol{x}) := \left(n_x^S(\boldsymbol{x}), n_y^S(\boldsymbol{x}), n_z^S(\boldsymbol{x})\right)$ stands for the unit outward normal vector to subdomain Ω^S. Notice that if $\boldsymbol{x} \in \Gamma^{\mathrm{C}}$, therefore, $\boldsymbol{n}^{\mathrm{A}}(\boldsymbol{x}) = -\boldsymbol{n}^{\mathrm{B}}(\boldsymbol{x})$.

The conjugate heat transfer problem consists in seeking temperature function $\phi^S(\boldsymbol{x})$ in subdomain Ω^S with governing partial differential equation given as

$$\nabla \cdot \left(\boldsymbol{u}^S \phi^S - \kappa^S \nabla \phi^S\right) = f^S, \quad \text{in } \Omega^S, \tag{5.1}$$

where $\boldsymbol{u}^S := \boldsymbol{u}^S(\boldsymbol{x}) := \left(u_x^S(\boldsymbol{x}), u_y^S(\boldsymbol{x}), u_z^S(\boldsymbol{x})\right)$ is the velocity vector function multiplied by the heat capacity and density of the associated material, $\kappa^S := \kappa^S(\boldsymbol{x})$ is the thermal conductivity function, and $f^S := f^S(\boldsymbol{x})$ is the heat source function (a negative value implies a heat sink) defined in subdomain Ω^S. All given and unknown functions are assumed to be regular and bounded in the associated subdomains, although the velocity, thermal conductivity, and heat source can be discontinuous on the interface. Moreover, function $\boldsymbol{u}^S(\boldsymbol{x})$ is tangent to interface Γ^{C}, that is, $\boldsymbol{u}^S(\boldsymbol{x}) \cdot \boldsymbol{n}^S(\boldsymbol{x}) = 0$, such that there is no mass transfer from one subdomain to the other.

Remark 20 The term regular is used to indicate that the function belongs to spaces C^{k+1}, with $k \geq d$, when polynomial reconstruction of degree d is used to achieve a $(d+1)$-order accurate scheme. In other words, the function is continuous and has, at least, continuous first $k+1$ derivatives.

The model problem is complemented with Robin boundary conditions on boundary subsets $\Gamma^{\mathrm{R},S}$, given as

$$\alpha^{\mathrm{R},S} \phi^S + \beta^{\mathrm{R},S}\left(-\kappa^S \nabla \phi^S \cdot \boldsymbol{n}^S\right) = g^{\mathrm{R},S}, \quad \text{on } \Gamma^{\mathrm{R},S}, \tag{5.2}$$

where $\alpha^{\mathrm{R},S} := \alpha^{\mathrm{R},S}(\boldsymbol{x})$, $\beta^{\mathrm{R},S} := \beta^{\mathrm{R},S}(\boldsymbol{x})$, and $g^{\mathrm{R},S} := g^{\mathrm{R},S}(\boldsymbol{x})$ are given regular and bounded coefficient functions. Notice that, the Robin boundary condition provides a general approach to prescribe the Dirichlet boundary condition setting coefficient functions $\alpha^{\mathrm{R},S}(\boldsymbol{x}) \neq 0$ and $\beta^{\mathrm{R},S}(\boldsymbol{x}) = 0$ and the Neumann boundary condition assigning coefficient functions $\alpha^{\mathrm{R},S}(\boldsymbol{x}) = 0$ and $\beta^{\mathrm{R},S}(\boldsymbol{x}) \neq 0$. Moreover, having a general approach to prescribe boundary conditions, a unique numerical treatment is provided simplifying both the proposed method and the algorithm.

The model problem is also complemented with the continuity interface conditions (or homogeneous jump interface conditions) on interface subset Γ^{C}, given as

$$\left.\begin{aligned} &\phi^{\mathrm{A}} = \phi^{\mathrm{B}}, && (5.3)\\ &-\kappa^{\mathrm{A}}\nabla\phi^{\mathrm{A}}\cdot\boldsymbol{n}^{\mathrm{A}} - \kappa^{\mathrm{B}}\nabla\phi^{\mathrm{B}}\cdot\boldsymbol{n}^{\mathrm{B}} = 0, && (5.4)\end{aligned}\right\} \quad \text{on } \Gamma^{\mathrm{C}},$$

which imposes the continuity of the temperature and the conservation of the conductive heat flux. Notice that, for adjacent materials with different thermal conductivities, the normal derivative of the temperature on the interface can be discontinuous due to the conservation of the conductive heat flux.

5.2.2 Partitioned problem model

Following the approach in Costa et al., 2019 [15], the procedure to solve the conjugate heat transfer problem consists in a partitioning method (called the Dirichlet-Neumann method) that replaces the interface conditions with Dirichlet and Neumann boundary conditions on the interface. Then, a subproblem in each subdomain is formulated with only boundary conditions, and the thermal coupling between subproblems consists in providing the appropriate values to the boundary conditions such that the heat transfer is recovered.

The boundaries of subdomains Ω^S correspond to some partition of the boundary of domain Ω and the interface Γ^{C}. Therefore, for notation convenience, $\Gamma^{F,S}$, $F \in \{\mathrm{R},\mathrm{C}\}$, denotes the subsets of the partitioned boundaries of subdomains Ω^S such that $\Gamma^S = \Gamma^{\mathrm{R},S} \cup \Gamma^{\mathrm{C},S}$. In that way, boundary subset $\Gamma^{\mathrm{C},S}$ of subdomains Ω^S correspond to interface subset Γ^{C} of domain Ω and, therefore, $\Gamma^{\mathrm{C,A}} = \Gamma^{\mathrm{C,B}} = \Gamma^{\mathrm{C}}$.

Subproblem P^S consists in seeking temperature function $\phi^S(\boldsymbol{x})$ in subdomain Ω^S with governing partial differential equation and boundary conditions given as

$$\mathrm{P}^S : \begin{cases} \nabla\cdot\left(\boldsymbol{u}^S\phi^S - \kappa^S\nabla\phi^S\right) = f^S, & \text{in } \Omega^S, \quad (5.5)\\ \alpha^{F,S}\phi^S + \beta^{F,S}\left(-\kappa^S\nabla\phi^S\cdot\boldsymbol{n}^S\right) = g^{F,S}, & \text{on } \Gamma^{F,S}, \quad (5.6)\end{cases}$$

where $\alpha^{F,S} := \alpha^{F,S}(\boldsymbol{x})$, $\beta^{F,S} := \beta^{F,S}(\boldsymbol{x})$, and $g^{F,S} := g^{F,S}(\boldsymbol{x})$ are regular and bounded functions on boundary subset $\Gamma^{F,S}$. On boundary subsets $\Gamma^{\mathrm{R},S}$, functions $\alpha^{\mathrm{R},S} := \alpha^{\mathrm{R},S}(\boldsymbol{x})$, $\beta^{\mathrm{R},S} := \beta^{\mathrm{R},S}(\boldsymbol{x})$, and $g^{\mathrm{R},S} := g^{\mathrm{R},S}(\boldsymbol{x})$ are prescribed from the conjugate problem model. On boundary subsets $\Gamma^{\mathrm{C},S}$, functions $\alpha^{\mathrm{C},S} := \alpha^{\mathrm{C},S}(\boldsymbol{x})$, $\beta^{\mathrm{C},S} := \beta^{\mathrm{C},S}(\boldsymbol{x})$, and $g^{\mathrm{C},S} := g^{\mathrm{C},S}(\boldsymbol{x})$ were not prescribe and, therefore, are unknown in the partitioned problem model.

To recover the appropriate thermal coupling between the subproblems and following the approach in Costa et. al, 2019 [15], the continuity interface conditions (5.3) and (5.4) prescribed on interface subset Γ^{C} are replaced with a Dirichlet boundary condition for subproblem P^{A} and a

Neumann boundary condition for subproblem P^{B}. However, since a Robin boundary condition is prescribed on the entire boundary of the subdomains for a general approach, coefficient functions on boundary subset $\Gamma^{\mathrm{C,A}}$ are assigned as $\alpha^{\mathrm{C,A}}(\boldsymbol{x}) = 1$ and $\beta^{\mathrm{C,A}}(\boldsymbol{x}) = 0$ and on boundary subset $\Gamma^{\mathrm{C,B}}$ are assigned as $\alpha^{\mathrm{C,B}}(\boldsymbol{x}) = 0$ and $\beta^{\mathrm{C,B}}(\boldsymbol{x}) = 1$. On the other hand, associated boundary condition functions $g^{\mathrm{C,A}}(\boldsymbol{x})$ and $g^{\mathrm{C,B}}(\boldsymbol{x})$ are given as

$$g^{\mathrm{C,A}} := \phi^{\mathrm{B}}, \qquad \text{on } \Gamma^{\mathrm{C,A}}, \tag{5.7}$$

$$g^{\mathrm{C,B}} := -\kappa^{\mathrm{A}} \nabla \phi^{\mathrm{A}} \cdot \boldsymbol{n}^{\mathrm{B}}, \quad \text{on } \Gamma^{\mathrm{C,B}}, \tag{5.8}$$

which depend on temperature functions $\phi^{\mathrm{B}}(\boldsymbol{x})$ and $\phi^{\mathrm{A}}(\boldsymbol{x})$, respectively. Since the temperature functions are also unknown, a numerical treatment to determine the unknown boundary condition functions for the thermal coupling is required.

The benefit of the partitioning approach is that each subproblem consists in seeking temperature function $\phi^{S}(\boldsymbol{x})$ in subdomain Ω^{S} with only boundary conditions. Therefore, each subproblem is discretized separately, whereas, boundary condition functions $g^{\mathrm{C,A}}(\boldsymbol{x})$ and $g^{\mathrm{C,B}}(\boldsymbol{x})$ on the interface are considered for the data transfer and require a specific treatment to recover the appropriate thermal coupling. Boundary condition functions $g^{\mathrm{C,A}}(\boldsymbol{x})$ and $g^{\mathrm{C,B}}(\boldsymbol{x})$ are assigned such that the solution of the partitioned problem satisfies the continuity interface conditions and, therefore, corresponds to the solution of the conjugate heat transfer problem.

Remark 21 The Dirichlet-Neumann method applied for the continuity interface conditions yields two asymmetric subproblems with a Dirichlet boundary condition on boundary subset $\Gamma^{C,A}$ and a Neumann boundary condition on boundary subset $\Gamma^{C,B}$. The opposite situation, a Neumann boundary condition on boundary subset $\Gamma^{C,A}$ and a Dirichlet boundary condition on boundary subset $\Gamma^{C,B}$, is equally possible and equivalent in terms of thermal coupling but may have accuracy and robustness implications. Although both partitioning configurations are addressed in the numerical benchmark, only the former is used for the method presentation.

5.3 Finite volume discretization

The domain of the conjugate heat transfer problem is discretized with polyhedral meshes, conformal on the interface, although the boundary value subproblems in the partitioned problem model are discretized separately. The finite volume method is used for the generic discretization of the subproblems, where an extension of the method proposed in Costa et al., 2019 [15], to three-dimensional domains and Robin boundary conditions was developed to provide a very high-order of convergence.

5.3.1 Polyhedral mesh

A general polyhedral mesh denoted as $\mathcal{M}^S$ discretizes each subdomain Ω^S and consists of n^S non-overlapping convex polyhedral cells (tetrahedra, prisms, hexahedra, etc.). Cells are denoted as c_i with $i \in \mathcal{I}^{\mathrm{A}} = \{1, \ldots, n^{\mathrm{A}}\}$ for cells belonging to mesh $\mathcal{M}^{\mathrm{A}}$ and $i \in \mathcal{I}^{\mathrm{B}} = \{n^{\mathrm{A}} + 1, \ldots, n^{\mathrm{A}} + n^{\mathrm{B}}\}$ for cells belonging to mesh $\mathcal{M}^{\mathrm{B}}$ and, therefore, the index of each cell is unique.

Inner faces are denoted as f_{ij}, $i, j \in \mathcal{I}^S$, and correspond to the common faces between two adjacent cells c_i and c_j, $j \neq i$, belonging to mesh $\mathcal{M}^S$ and, therefore, $f_{ij} = c_i \cap c_j$. Boundary faces are denoted as f_{iF}, $i \in \mathcal{I}^{F,S}$, and correspond to the faces of cells c_i belonging to mesh $\mathcal{M}^S$ that approximate boundary subsets $\Gamma^{F,S}$ (for the sake of simplicity, each cell has at most one boundary face). Subset $\mathcal{I}^{F,S} \subset \mathcal{I}^S$ gathers the indices and $n^{F,S}$ is the number of the cells belonging to mesh $\mathcal{M}^S$ with a boundary face approximating boundary subset $\Gamma^{F,S}$.

The vertices of the boundary faces fall on the surfaces of the associated boundary subsets and, in particular, on boundary subsets $\Gamma^{\mathrm{C},S}$ are common to both meshes (conformal meshes on the interface). In that case, for each boundary face $f_{i\mathrm{C}}$ corresponding to a face of cell c_i, with $i \in \mathcal{I}^{\mathrm{C,A}}$ and belonging to mesh $\mathcal{M}^{\mathrm{A}}$, there is a counterpart boundary face $f_{j\mathrm{C}}$ corresponding to a face of cell c_j, with $j \in \mathcal{I}^{\mathrm{C,B}}$ and belonging to mesh $\mathcal{M}^{\mathrm{B}}$, such that faces $f_{i\mathrm{C}}$ and $f_{j\mathrm{C}}$ share the same vertices and are geometrically equivalent (same shape and location).

Table 5.1 introduces the geometric properties for the cells and faces and Figure 5.1 provides a schematic representation. Notice that inner face f_{ij} is also denoted as f_{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, $\boldsymbol{s}_{ij} = -\boldsymbol{s}_{ji}$. Similarly, for counterpart boundary faces $f_{i\mathrm{C}}$ and $f_{j\mathrm{C}}$, symmetric outward unit normal vectors are assigned, that is, $\boldsymbol{s}_{i\mathrm{C}} = -\boldsymbol{s}_{j\mathrm{C}}$.

Table 5.1: Notation and geometric properties for the cells and faces.

Mesh elements	Notation	Properties	Definition	Choice
Cells	$c_i \subset \Omega^S_\Delta$ $S \in \{\mathrm{A}, \mathrm{B}\}$	∂c_i	Boundary	
		$\lvert c_i \rvert$	Volume	
		$\boldsymbol{m}_i = (m_{i,x}, m_{i,y}, m_{i,z})$	Reference point (can be any point in c_i)	Centroid
		$\boldsymbol{q}_{i,q} = (q_{i,q,x}, q_{i,q,y}, q_{i,q,z})$	Quadrature points, $q = 1, \ldots, Q$	Gaussian
		$\mathcal{N}_i$	Indices of the adjacent cells and boundary subset	
Inner faces	$f_{ij} \subset \Omega^S_\Delta$ $S \in \{\mathrm{A}, \mathrm{B}\}$	$\lvert f_{ij} \rvert$	Area	
		$\boldsymbol{m}_{ij} = (m_{ij,x}, m_{ij,y}, m_{ij,z})$	Reference point (can be any point on f_{ij})	Centroid
		$\boldsymbol{q}_{ij,r} = (q_{ij,r,x}, q_{ij,r,y}, q_{ij,r,z})$	Quadrature points, $r = 1, \ldots, R$	Gaussian
		$\boldsymbol{s}_{ij} = (s_{ij,x}, s_{ij,y}, s_{ij,z})$	Outward unit normal vector from cell c_i to cell c_j	
Boundary faces	$f_{iF} \subset \Gamma^{F,S}_\Delta$ $F \in \{\mathrm{R}, \mathrm{C}\}$ $S \in \{\mathrm{A}, \mathrm{B}\}$	$\lvert f_{iF} \rvert$	Area	
		$\boldsymbol{m}_{iF} = (m_{iF,x}, m_{iF,y}, m_{iF,z})$	Reference point (can be any point on f_{iF})	Centroid
		$\boldsymbol{q}_{iF,r} = (q_{iF,r,x}, q_{iF,r,y}, q_{iF,r,z})$	Quadrature points, $r = 1, \ldots, R$	Gaussian
		$\boldsymbol{s}_{iF} = (s_{iF,x}, s_{iF,y}, s_{iF,z})$	Outward unit normal vector from c_i	

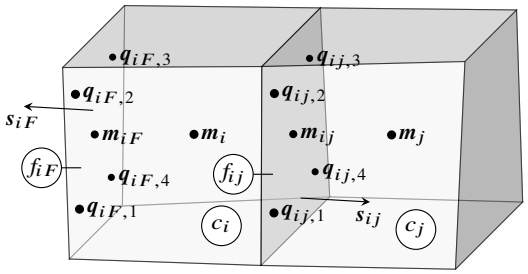


Figure 5.1: Notation and geometric properties for the cells and faces.

The term computational is used to distinguish the physical subdomains, boundaries, and boundary subsets, denoted as Ω^S, Γ^S, and $\Gamma^{F,S}$, respectively, from the corresponding discretized subdomains, boundaries, and boundary subsets, denoted as Ω^S_Δ, Γ^S_Δ, and $\Gamma^{F,S}_\Delta$, respectively, composed of the associated mesh elements. Since meshes $\mathcal{M}^S$ are composed of polyhedral cells, computational subdomains Ω^S_Δ do not fully overlap associated curved physical subdomains Ω^S. For fine enough meshes, a geometrical mismatch with magnitude order $\mathcal{O}(h^2)$ is then expected between the physical and the computational boundaries, where h is the characteristic mesh size. Such a geometrical mismatch leads to an accuracy deterioration and provides, at most, a second-order of convergence for the method if the appropriate treatment is not used.

5.3.2 Generic scheme

The generic discretization of subproblems P^S with the finite volume method consists in applying the divergence theorem to the integral form of Equation (5.5) in each cell c_i in associated computational subdomain Ω^S_Δ, given as

$$\int_{\partial c_i} \left(\boldsymbol{u}^S(\boldsymbol{x})\phi^S(\boldsymbol{x}) - \kappa^S(\boldsymbol{x})\nabla\phi^S(\boldsymbol{x}) \right) \cdot \boldsymbol{s}_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{c_i} f^S(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{5.9}$$

where ∂c_i stands for the cell boundary and $\boldsymbol{s}_i(\boldsymbol{x})$ is the associated outward unit normal vector.

A R-points quadrature rule with weights ζ_r, $r = 1, \ldots, R$, is used to approximate the integrals on the polygonal faces and a Q-points quadrature rule with weights ξ_q, $q = 1, \ldots, Q$, is used to approximate the integrals in the polyhedral cells. Gaussian quadrature rules are considered with weights that provide the approximate mean-values of the functions to evaluate. Given that, Equation (5.9) is rewritten in the discrete form in terms of numerical integrals, given as

$$\sum_{j \in \mathcal{N}_i} |f_{ij}| \left[\sum_{r=1}^{R} \zeta_r \left(C_{ij,r} + D_{ij,r} \right) \right] - f_i |c_i| = \mathcal{O}(h_i^\alpha), \tag{5.10}$$

where $h_i = |c_i|^{1/3}$ and α is the convergence order of the quadrature rule. Moreover, f_i stands for the approximate mean-value of source term function $f^S(\boldsymbol{x})$ in cell c_i and $C_{ij,r}$ and $D_{ij,r}$ stand for the physical convective and conductive heat fluxes, respectively, at the quadrature points of face f_{ij}, given

as

$$C_{ij,r} = \left(\boldsymbol{u}^S\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}\right)\phi^S\left(\boldsymbol{q}_{ij,r}\right), \tag{5.11}$$

$$D_{ij,r} = -\kappa^S\left(\boldsymbol{q}_{ij,r}\right)\nabla\phi^S\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}. \tag{5.12}$$

Notice that, for inner faces $f_{ij} = f_{ji}$, the conservation of the convective and conductive heat fluxes is intrinsically preserved within the finite volume method, that is, $C_{ij,r} = -C_{ji,r}$ and $D_{ij,r} = -D_{ji,r}$, since the associated quadrature points coincide and the associated unit normal vectors are symmetric, that is, $\boldsymbol{q}_{ij,r} = \boldsymbol{q}_{ji,r}$ and $\boldsymbol{s}_{ij} = -\boldsymbol{s}_{ji}$.

5.3.3 Polynomial reconstructions

Temperature functions are approximated with piecewise cell mean-values in the associated computational subdomain, such that for cell c_i in computational subdomain Ω_Δ^S, the associated cell mean-value is denoted as ϕ_i and defined as

$$\phi_i \approx \frac{1}{|c_i|}\int_{c_i}\phi^S(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{5.13}$$

and vector $\boldsymbol{\Phi}^S \in \mathbb{R}^{n^S}$ gathers the values for computational subdomain Ω_Δ^S.

A polynomial function of degree d, denoted as $\varphi(\boldsymbol{x})$, is written in a compacted form as

$$\varphi(\boldsymbol{x}) = \boldsymbol{\eta}\cdot\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}) = \sum_{\alpha=0}^{d}\sum_{\beta=0}^{d-\alpha}\sum_{\gamma=0}^{d-\alpha-\beta}\eta_{\alpha\beta\gamma}(x-m_x)^\alpha\left(y-m_y\right)^\beta(z-m_z)^\gamma, \tag{5.14}$$

where $\boldsymbol{p}_d(\boldsymbol{x})$ is a basis vector including all three-dimensional monomials up to degree d, $\boldsymbol{m} = \left(m_x, m_y, m_z\right)$ is a reference point, and vector $\boldsymbol{\eta} \in \mathbb{R}^n$ gathers the polynomial coefficients, with $n = (d+1)(d+2)(d+3)/6$.

The polynomial reconstruction method requires seeking, through a minimization procedure, polynomial coefficients vector $\boldsymbol{\eta}$ that provides the best approximation of polynomial function $\varphi(\boldsymbol{x})$ to the cell mean-values in the vicinity of a given mesh element (cell or face) and belonging to the same subdomain. The linear least-squares method is the standard minimization procedure and consists in minimizing the sum of the squared residuals that correspond to the difference between the cell mean-values of the polynomial function and the cell mean-values of the associated temperature function. For that purpose, consider the prototype weighted cost functional from $\mathbb{R}^n$ to $\mathbb{R}$, denoted as $F(\boldsymbol{\eta})$, given as

$$F(\boldsymbol{\eta}) = \sum_{k\in\mathcal{S}}\omega_k\left[\frac{1}{|c_k|}\int_{c_k}\varphi(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \phi_k\right]^2. \tag{5.15}$$

Without loss of accuracy, the integrals are evaluated numerically using quadrature rules of the same order as the polynomial degree. $\mathcal{S}$ is a stencil gathering the indices of s cells in the vicinity of the reference point ($s \approx 3.5n$ for the sake of robustness and stability), and ω_k, $k \in \mathcal{S}$, are positive weights given as $\omega_k = \omega(d_k) = 1 \Big/ \left((\sigma d_k)^\delta + 1\right)$, where $\delta, \sigma \in \mathbb{R}$ are given parameters and $d_k = |\boldsymbol{m}_k - \boldsymbol{m}|$. Vector $\boldsymbol{\omega} \in \mathbb{R}^s$ gathers the weights associated with the cells with an index in the stencil.

For unconstrained polynomial reconstructions, the minimization procedure consists in seeking unique vector $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes functional $F(\boldsymbol{\eta})$ in the least-squares sense, that is, $\widetilde{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} F(\boldsymbol{\eta})$. The optimal polynomial reconstruction is then written as $\widetilde{\varphi}(\boldsymbol{x}) = \widetilde{\boldsymbol{\eta}} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m})$. The normal equations method [16] is used to derive the unconstrained least-squares method and perform the associated minimization procedure in the present work.

For constrained polynomial reconstructions, functional $F(\boldsymbol{\eta})$ is minimized in the least-squares sense under a linear constraint to exactly fulfill. Denoting a linear constraint functional from $\mathbb{R}^n$ to $\mathbb{R}$ as $G(\boldsymbol{\eta})$, the minimization procedure consists in seeking unique vector $\widehat{\boldsymbol{\eta}} \in \mathbb{R}^n$ that minimizes functional $F(\boldsymbol{\eta})$ in the least-squares sense under constraint equation $G(\boldsymbol{\eta}) = 0$, that is, $\widehat{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} F(\boldsymbol{\eta})$ subject to $G(\boldsymbol{\eta}) = 0$. The optimal polynomial reconstruction is written as $\widehat{\varphi}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m})$. The so-called linearly constrained Lagrange multipliers method [17] is used to derive the constrained least-squares method and perform the associated minimization procedure in the present work.

Unconstrained polynomial reconstructions are associated with the inner faces, whereas constrained polynomial reconstructions are associated with the boundary faces and the cells, as follows:

- Unconstrained polynomial reconstructions $\widetilde{\varphi}_{ij}(\boldsymbol{x}) = \widetilde{\boldsymbol{\eta}}_{ij} \cdot \boldsymbol{p}_d\left(\boldsymbol{x} - \boldsymbol{m}_{ij}\right)$ are computed for each inner face f_{ij} in computational subdomain Ω_Δ^S. For the minimization procedure, the parameters are assigned as $\boldsymbol{\eta} \coloneqq \boldsymbol{\eta}_{ij}$, $\boldsymbol{m} \coloneqq \boldsymbol{m}_{ij}$, $\mathcal{S} \coloneqq \mathcal{S}_{ij}$, and $\boldsymbol{\omega} \coloneqq \boldsymbol{\omega}_{ij}$.

- Constrained polynomial reconstructions $\widehat{\varphi}_i(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_i \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_i)$ are computed for each cell c_i in computational subdomain Ω_Δ^S. For the minimization procedure, the parameters are assigned as $\boldsymbol{\eta} \coloneqq \boldsymbol{\eta}_i$, $\boldsymbol{m} \coloneqq \boldsymbol{m}_i$, $\mathcal{S} \coloneqq \mathcal{S}_i$, and $\boldsymbol{\omega} \coloneqq \boldsymbol{\omega}_i$, and the linear constraint consists in preserving mean-value ϕ_i in cell c_i. Therefore, constraint functional $G(\boldsymbol{\eta}) \coloneqq G_i\left(\boldsymbol{\eta}_i\right)$ is given as

$$G_i\left(\boldsymbol{\eta}_i\right) = \frac{1}{|c_i|} \int_{c_i} \varphi_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \phi_i. \tag{5.16}$$

- Constrained polynomial reconstructions $\widehat{\varphi}_{iF}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_{iF} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{iF})$ are assigned for each boundary face f_{iF} on computational boundary subset $\Omega_\Delta^{F,S}$. For the minimization procedure, the parameters are assigned as $\boldsymbol{\eta} \coloneqq \boldsymbol{\eta}_{iF}$, $\boldsymbol{m} \coloneqq \boldsymbol{m}_{iF}$, $\mathcal{S} \coloneqq \mathcal{S}_{iF}$, and $\boldsymbol{\omega} \coloneqq \boldsymbol{\omega}_{iF}$, and the linear constraint consists in fulfilling the prescribed boundary condition function with point-value g_{iF} at given collocation point $\boldsymbol{p}_{iF} = \left(p_{iF,x}, p_{iF,y}, p_{iF,z}\right)$ with unit vector $\boldsymbol{v}_{iF} = \left(v_{iF,x}, v_{iF,y}, v_{iF,z}\right)$.

Therefore, constraint functional $G(\boldsymbol{\eta}) := G_{iF}\left(\boldsymbol{\eta}_{iF}\right)$ is given as

$$G_{iF}\left(\boldsymbol{\eta}_{iF}\right) = \alpha^{F,S}(\boldsymbol{p}_{iF})\varphi_{iF}(\boldsymbol{p}_{iF}) + \beta^{F,S}(\boldsymbol{p}_{iF})\Big(-\kappa^{S}(\boldsymbol{p}_{iF})\nabla\varphi_{iF}(\boldsymbol{p}_{iF})\cdot\boldsymbol{v}_{iF}\Big) - g_{iF}. \quad (5.17)$$

The ROD method is used to avoid the deterioration of the accuracy and convergence order of the method and consists in defining collocation point $\boldsymbol{p}_{iF} := \boldsymbol{b}_{iF}$ and unit vector $\boldsymbol{v}_{iF} := \boldsymbol{n}_{iF} = \boldsymbol{n}^{S}(\boldsymbol{b}_{iF})$ belonging to associated curved physical boundary subset $\Gamma^{F,S}$. Collocation point $\boldsymbol{b}_{iF}$ derived from the projection of face centroid $\boldsymbol{m}_{iF}$ onto the associated physical boundary subset.

5.3.4 Very high-order accurate scheme

The very high-order accurate scheme consists in providing numerical approximations to physical convective and conductive heat fluxes $C_{ij,r}$ and $D_{ij,r}$, respectively, from the values and gradients of the polynomial reconstructions. Consider the convention $[a]^{+} = \max(a,0)$ and $[a]^{-} = \min(a,0)$ as an upwind indicator. The numerical convective and conductive heat fluxes are given as follows:

- For each inner face f_{ij} in computational subdomain Ω_{Δ}^{S}, numerical convective and conductive heat fluxes, denoted as $\mathcal{C}_{ij,r}$ and $\mathcal{D}_{ij,r}$, respectively, at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1,\ldots,R$, are given as

$$\mathcal{C}_{ij,r} = \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}\right]^{+}\widehat{\varphi}_{i}\left(\boldsymbol{q}_{ij,r}\right) + \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}\right]^{-}\widehat{\varphi}_{j}\left(\boldsymbol{q}_{ij,r}\right), \quad (5.18)$$

$$\mathcal{D}_{ij,r} = -\kappa^{S}\left(\boldsymbol{q}_{ij,r}\right)\nabla\widetilde{\varphi}_{ij}\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}. \quad (5.19)$$

- For each boundary face f_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$, numerical convective and conductive heat fluxes, denoted as $\mathcal{C}_{iF,r}$ and $\mathcal{D}_{iF,r}$, respectively, at quadrature points $\boldsymbol{q}_{iF,r}$, $r = 1,\ldots,R$, are given as

$$\mathcal{C}_{iF,r} = \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}\right]^{+}\widehat{\varphi}_{i}\left(\boldsymbol{q}_{iF,r}\right) + \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}\right]^{-}\widehat{\varphi}_{iF}\left(\boldsymbol{q}_{iF,r}\right), \quad (5.20)$$

$$\mathcal{D}_{iF,r} = -\kappa^{S}\left(\boldsymbol{q}_{iF,r}\right)\nabla\widehat{\varphi}_{iF}\left(\boldsymbol{q}_{iF,r}\right)\cdot\boldsymbol{s}_{iF}. \quad (5.21)$$

Notice that only polynomial reconstructions computed for the subproblem associated with the inner or boundary faces are used to determine the numerical approximations of the physical heat fluxes. Moreover, the prescribed boundary conditions are taken into account to compute the constrained polynomial reconstructions associated with the boundary faces, and no explicit reference appears in the discretization scheme, which handles only two situations, inner or boundary faces.

5.3.5 Boundary variables

In Equation (5.17), point-value g_{iF} stands for the value at collocation point $\boldsymbol{p}_{iF}$ of the Robin boundary condition function to fulfill, which is determined as follows:

- For each boundary face $f_{i\mathrm{R}}$ on computational boundary subset $\Gamma^{\mathrm{R},S}$, point-value $g_{iF} \coloneqq g_{i\mathrm{R}}$ is given as $g_{i\mathrm{R}} = g^{\mathrm{R},S}(\boldsymbol{p}_{i\mathrm{R}})$ and is explicitly determined from boundary condition function $g^{\mathrm{R},S}(\boldsymbol{x})$.

- For each boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,A}}$, point-value $g_{iF} \coloneqq g_{i\mathrm{C}}$ is given as $g_{i\mathrm{C}} = g^{\mathrm{C,A}}\left(\boldsymbol{p}_{i\mathrm{C}}\right)$ and can not be explicitly determined since boundary condition function $g^{\mathrm{C,A}}(\boldsymbol{x})$ is implicitly defined in Equation (5.7). Therefore, point-value $g_{i\mathrm{C}}$ is implicitly determined with the evaluation of polynomial reconstruction $\widehat{\varphi}_j(\boldsymbol{x})$, associated to the counterpart boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma^{\mathrm{C,B}}$, at collocation point $\boldsymbol{p}_{i\mathrm{C}}$, given as

$$g_{i\mathrm{C}} = \widehat{\varphi}_j\left(\boldsymbol{p}_{i\mathrm{C}}\right). \tag{5.22}$$

  Notice that $\alpha^{\mathrm{C,A}}(\boldsymbol{x}) = 1$ and $\beta^{\mathrm{C,A}}(\boldsymbol{x}) = 0$ and the linear constraint for the minimization procedure corresponds to the Dirichlet boundary condition.

- For each boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$, point-value $g_{iF} \coloneqq g_{j\mathrm{C}}$ is given as $g_{j\mathrm{C}} = g^{\mathrm{C,B}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)$ and can not be explicitly determined since boundary condition function $g^{\mathrm{C,B}}(\boldsymbol{x})$ is implicitly defined in Equation (5.8). Therefore, point-value $g_{j\mathrm{C}}$ is implicitly determined with the evaluation of polynomial reconstruction $\widehat{\varphi}_{i\mathrm{C}}(\boldsymbol{x})$, associated to the counterpart boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma^{\mathrm{C,A}}$, at collocation point $\boldsymbol{p}_{j\mathrm{C}}$ with associated unit normal vector $\boldsymbol{v}_{j\mathrm{C}}$, given as

$$g_{j\mathrm{C}} = -\kappa^{\mathrm{A}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\nabla\widehat{\varphi}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\cdot\boldsymbol{v}_{j\mathrm{C}}. \tag{5.23}$$

  Notice that $\alpha^{\mathrm{C,B}}(\boldsymbol{x}) = 0$ and $\beta^{\mathrm{C,B}}(\boldsymbol{x}) = 1$ and the linear constraint for the minimization procedure corresponds to the Neumann boundary condition.

5.3.6 Residual operators

The cell mean-values of temperature functions $\phi^S(\boldsymbol{x})$ are gathered in vectors $\boldsymbol{\Phi}^S \in \mathbb{R}^{n^S}$ and the point-values of boundary condition functions $g^{F,S}(\boldsymbol{x})$ are gathered in vectors $\boldsymbol{\Psi}^{F,S} \in \mathbb{R}^{n^{F,S}}$, that is, $\boldsymbol{\Phi}^S = [\phi_i]_{i\in\mathcal{I}^S}$ and $\boldsymbol{\Psi}^{F,S} = [g_{iF}]_{i\in\mathcal{I}^{F,S}}$. The equations derived from proposed method, that is, the discretization of the subproblems and the numerical treatment of the thermal coupling, are rewritten in terms of the residual of affine operators, as follows:

- For each cell c_i in computational subdomain Ω_{Δ}^S, Equation (5.10) derived from the discretization

of the subproblems is rewritten with the numerical heat fluxes as $\mathcal{R}_i = 0$ with residual given as

$$\mathcal{R}_i = \sum_{j \in \mathcal{N}_i} |f_{ij}| \left[\sum_{r=1}^{R} \zeta_r \left(\mathcal{C}_{ij,r} + \mathcal{D}_{ij,r} \right) \right] - f_i |c_i|. \tag{5.24}$$

The residuals defined in Equation (5.24) for cells c_i in computational subdomain Ω_Δ^S with only inner faces are gathered in the vector form with affine residual operator $\mathcal{R}^S\left(\boldsymbol{\Phi}^S\right)$ from $\mathbb{R}^{n^S}$ to $\mathbb{R}^{n^S - n^{\mathrm{R},S} - n^{\mathrm{C},S}}$. The residuals defined in Equation (5.24) for cells c_i in computational subdomain Ω_Δ^S with a boundary face f_{iF} on computational boundary subset $\Gamma_\Delta^{F,S}$ are gathered in the vector form with affine residual operator $\mathcal{R}^{F,S}\left(\boldsymbol{\Phi}^S, \boldsymbol{\Psi}^{F,S}\right)$ from $\mathbb{R}^{n^S + n^{F,S}}$ to $\mathbb{R}^{n^{F,S}}$.

- For each boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{C,A}}$ and counterpart boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{C,B}}$, Equations (5.22) and (5.23) derived from the numerical treatment of the thermal coupling are rewritten as $\mathcal{G}_{i\mathrm{C}} = 0$ and $\mathcal{G}_{j\mathrm{C}} = 0$, respectively, with residuals given as

$$\mathcal{G}_{i\mathrm{C}} = \widehat{\varphi}_j\left(\boldsymbol{p}_{i\mathrm{C}}\right) - g_{i\mathrm{C}}, \tag{5.25}$$

$$\mathcal{G}_{j\mathrm{C}} = -\kappa^{\mathrm{A}}\left(\boldsymbol{p}_{j\mathrm{C}}\right) \nabla \widehat{\varphi}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right) \cdot \boldsymbol{\nu}_{j\mathrm{C}} - g_{j\mathrm{C}}. \tag{5.26}$$

The residuals defined in Equation (5.25) for boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{C,A}}$ are gathered in the vector form with affine residual operator $\mathcal{G}^{\mathrm{C,A}}\left(\boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right)$ from $\mathbb{R}^{n^{\mathrm{C,A}} + n^{\mathrm{B}}}$ to $\mathbb{R}^{n^{\mathrm{C,A}}}$. The residuals defined in Equation (5.26) for counterpart boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{C,B}}$ are gathered in the vector form with affine residual operator $\mathcal{G}^{\mathrm{C,B}}\left(\boldsymbol{\Psi}^{\mathrm{C,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right)$ from $\mathbb{R}^{n^{\mathrm{C,B}} + n^{\mathrm{A}} + n^{\mathrm{C,A}}}$ to $\mathbb{R}^{n^{\mathrm{C,B}}}$.

5.3.7 Implicit formulation

The cell mean-value of temperature functions $\phi^S(\boldsymbol{x})$ in vectors $\boldsymbol{\Phi}^S$ are unknown and correspond to the internal variables for the proposed method while the point-values of boundary condition functions $g^{\mathrm{C},S}(\boldsymbol{x})$ in vectors $\boldsymbol{\Psi}^{\mathrm{C},S}$ are also unknown but correspond to the boundary variables for the proposed method. On the other side, boundary conditions functions $g^{\mathrm{R},S}(\boldsymbol{x})$ are given and, therefore, vectors $\boldsymbol{\Psi}^{\mathrm{R},S}$ only gather provided point-values. Therefore, the number of degrees of freedom to determine is $DOF = n^{\mathrm{A}} + n^{\mathrm{B}} + n^{\mathrm{C,A}} + n^{\mathrm{C,B}}$. The internal and boundary variables for the proposed method are gathered in global vector $\boldsymbol{\Phi} \in \mathbb{R}^{DOF}$, given as $\boldsymbol{\Phi} = \left[\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Phi}^{\mathrm{B}}, \boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Psi}^{\mathrm{C,B}}\right]^{\mathrm{T}}$.

The implicit system of linear equations for the solution of the partitioned problem is formed gathering residual operators (5.24) and (5.26) in vector form in affine global residual operator $\mathcal{R}(\boldsymbol{\Phi})$

from $\mathbb{R}^{DOF}$ to $\mathbb{R}^{DOF}$, written as

$$\mathcal{R}(\boldsymbol{\Phi}) = \begin{bmatrix} \mathcal{R}^{\mathrm{A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}\right) \\ \mathcal{R}^{\mathrm{R,A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{R,A}}\right) \\ \mathcal{R}^{\mathrm{C,A}}\left(\boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right) \\ \mathcal{R}^{\mathrm{B}}\left(\boldsymbol{\Phi}^{\mathrm{B}}\right) \\ \mathcal{R}^{\mathrm{R,B}}\left(\boldsymbol{\Phi}^{\mathrm{B}}, \boldsymbol{\Psi}^{\mathrm{C,B}}\right) \\ \mathcal{R}^{\mathrm{C,B}}\left(\boldsymbol{\Phi}^{\mathrm{B}}, \boldsymbol{\Psi}^{\mathrm{C,B}}\right) \\ \mathcal{G}^{\mathrm{C,A}}\left(\boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) \\ \mathcal{G}^{\mathrm{C,B}}\left(\boldsymbol{\Psi}^{\mathrm{C,B}}, \boldsymbol{\Phi}^{\mathrm{A}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right) \end{bmatrix}. \tag{5.27}$$

The solution of the implicit affine system given as $\mathcal{R}(\boldsymbol{\Phi}) = \boldsymbol{0}$ provides vector $\boldsymbol{\Phi}^* \in \mathbb{R}^{DOF}$ that corresponds to the approximate solution of the conjugate heat transfer problem in terms of internal and boundary variables. An iterative free-matrix algorithm, such as the GMRES method, supplemented with a preconditioning matrix, is used to solve the implicit system of linear equations. A static condensation method provides an implicit system of linear equations only in terms of internal variables, as implemented in Costa et al., 2019 [15].

5.4 Implementation improvements

Most of the execution time of a simulation spent on the computation on the linear system solution rather than on the initialization procedures, such as the generation of mesh, computation of quadrature rules and preconditioning matrices, among others. The implicit formulation of the proposed method requires an iterative free-matrix algorithm where each iteration consists in providing an input vector $\boldsymbol{\Phi}$ (in the case of the GMRES method it corresponds to the last vector in the Krylov subspace) and computing the global residual operator $\mathcal{R}(\boldsymbol{\Phi})$. A simple implementation of the proposed method in an implicit formulation through a global residual operator comprises the following steps:

- Step 1 – compute the coefficients vector of the polynomial reconstructions for each cell, inner face, and boundary face.
- Step 2 – determine the values and gradients of the polynomial reconstructions at the quadrature points on the faces and the collocation points.
- Step 3 – evaluate the numerical heat fluxes on each inner and boundary face and the boundary variables at each collocation point.

- Step 4 – compute the residual for the cells and the relation of the thermal coupling for each boundary variable.

Such simple implementation is not computationally efficient since several parameters and matrices in the proposed method are computed repeatedly without output variation at each iteration of the GMRES method. An optimized implementation is proposed deriving equations in each of these steps that perform the same operations and provide the same result with a reduced computational cost and, therefore, improving the performance of the proposed method. Moreover, an explicit formulation of the proposed method is derived where a sparse matrix of coefficients and a right-hand side vector replace the global residual operator.

5.4.1 Step 1

An optimization of step 1 is proposed to reduce the computational cost of computing the coefficients vector of the polynomial reconstructions for each cell, inner face, and boundary face.

5.4.1.1 Optimization

Several types of polynomial reconstructions for the finite volume scheme and the numerical treatment of the thermal coupling are required, namely, unconstrained polynomial reconstruction $\widetilde{\varphi}_{ij}(\boldsymbol{x})$ for each inner face f_{ij} in computational subdomain Ω_{Δ}^{S}, constrained polynomial reconstruction $\widehat{\varphi}_{i}(\boldsymbol{x})$ for each cell c_i in computational subdomain Ω_{Δ}^{S}, and constrained polynomial reconstruction $\widehat{\varphi}_{iF}(\boldsymbol{x})$ for each boundary face f_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$, given as

$$\widetilde{\varphi}_{ij}(\boldsymbol{x}) = \widetilde{\boldsymbol{\eta}}_{ij} \cdot \boldsymbol{p}_d\left(\boldsymbol{x} - \boldsymbol{m}_{ij}\right), \tag{5.28}$$

$$\widehat{\varphi}_{i}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_{i} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{i}), \tag{5.29}$$

$$\widehat{\varphi}_{iF}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_{iF} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{iF}). \tag{5.30}$$

Polynomial coefficients vectors $\widehat{\boldsymbol{\eta}}_{i}$, $\widetilde{\boldsymbol{\eta}}_{ij}$, and $\widehat{\boldsymbol{\eta}}_{iF}$ are computed performing the associated minimization procedure (unconstrained or constrained) at each iteration of the iterative free-matrix algorithm. However, the least-squares method reads to a linear method system where the matrices do not depend on the solution and can be computed at an earlier stage (see Appendix A). Polynomial coefficients vectors $\widehat{\boldsymbol{\eta}}_{i}$, $\widetilde{\boldsymbol{\eta}}_{ij}$, and $\widehat{\boldsymbol{\eta}}_{iF}$ can be computed respectively as

$$\widetilde{\boldsymbol{\eta}}_{ij} = \widetilde{\boldsymbol{M}}_{ij}\boldsymbol{b}_{ij}, \tag{5.31}$$

$$\widehat{\boldsymbol{\eta}}_{i} = \widehat{\boldsymbol{M}}_{i}\boldsymbol{b}_{i}, \tag{5.32}$$

$$\widehat{\boldsymbol{\eta}}_{iF} = \widehat{\boldsymbol{M}}_{iF}\boldsymbol{b}_{iF} + g_{iF}\widehat{\boldsymbol{n}}_{iF}. \tag{5.33}$$

5.4.1.2 Implementation

Matrices $\widetilde{\boldsymbol{M}}_{ij}, \widehat{\boldsymbol{M}}_{i}, \widehat{\boldsymbol{M}}_{iF} \in \mathbb{R}^{n \times s}$ and vector $\widehat{\boldsymbol{n}}_{iF} \in \mathbb{R}^{n}$ depend on geometrical parameters, namely the mesh, the polynomial degree, and the linear constraint in the latter cases. On the other side, vectors $\boldsymbol{b}_{ij} \in \mathbb{R}^{s}$, $\boldsymbol{b}_{i} \in \mathbb{R}^{s+1}$, and $\boldsymbol{b}_{iF} \in \mathbb{R}^{s}$ gather the cell mean-values (internal variables) associated to the cells with an index in stencils $\mathcal{S}_{ij}$, $\mathcal{S}_{i}$, and $\mathcal{S}_{iF}$, respectively, and g_{iF} is a boundary variable. Therefore, matrices $\widetilde{\boldsymbol{M}}_{ij}$, $\widehat{\boldsymbol{M}}_{i}$, and $\widehat{\boldsymbol{M}}_{iF}$ and vector $\widehat{\boldsymbol{n}}_{iF}$ are computed in advance at the pre-processing stage, stored in memory, and used in step 1.

The computational cost of the additional pre-processing stage is low, when compared with the cost of the iterative free-matrix method to solve the system of linear equations. The cost becomes negligible, especially when the number of iterations is high, as in slow convergence problems or even in non-linear and unsteady problems. On the other hand, the computational cost of the global residual operator drastically decreases since it avoids performing the minimization procedure for each polynomial reconstruction at every iteration. Indeed, the least-squares method is computationally heavy due to the cumbersome operations required to compute matrices $\widetilde{\boldsymbol{M}}_{ij}$, $\widehat{\boldsymbol{M}}_{i}$, and $\widehat{\boldsymbol{M}}_{iF}$ and vector $\widehat{\boldsymbol{n}}_{iF}$, when compared with the matrix-vector products required to compute polynomial coefficients vectors $\widehat{\boldsymbol{\eta}}_{i}$, $\widetilde{\boldsymbol{\eta}}_{ij}$, and $\widehat{\boldsymbol{\eta}}_{iF}$ with Equations (5.31) to (5.33). Therefore, the overall performance of the proposed method is improved, reducing the execution time of the simulation.

5.4.1.3 Computational complexity

The computational complexity of the algorithm for the global residual operator, after optimization of step 1, is estimated to provide an insight on the computational gains with the subsequent optimizations. At this stage, the most critical part of the algorithm concerns the calculation of the polynomial coefficients vectors with Equations (5.31) to (5.33), for which the corresponding number of floating-point operations and computational complexity are reported in Table 5.2. The computational complexity is estimated in terms of n, assuming $s = 3n$ and taking only the highest polynomial term.

Table 5.2: Floating-point operations and computational complexity for the calculation of the polynomial coefficients vectors.

Equation	Product dimensions	Floating-point operations	Computational complexity
(5.31)	$(n \times s) \times s$	$n(2s-1)$	$\mathcal{O}(6n^2)$
(5.32)	$(n \times (s+1)) \times (s+1)$	$n(2s+1)$	$\mathcal{O}(6n^2)$
(5.33)	$(n \times s) \times s,\ 1 \times n$	$n(2s)$	$\mathcal{O}(6n^2)$

After the calculation of the polynomial coefficients vectors, the numerical convective and conductive fluxes on the faces are determined. For that, the polynomial basis vector and the associated Jacobian matrix are necessary at each quadrature point, which can be determined with a minimum

number of $3(n-4)$ and $9(n-4)$ floating-point operations, respectively, using the Horner's rule. Then, the values and gradients of the polynomial reconstructions are obtained multiplying the polynomial basis vector and the associated Jacobian matrix by the corresponding polynomial coefficients vectors, which demand $2n-1$ and $3(2n-1)$ floating-point operations, respectively. Finally, the computational cost of the residuals assembling for the cells can be neglected since it takes considerably fewer floating-point operations than the remaining algorithm. For the same reason, the evaluation of the polynomial reconstructions at the collocation points is not considered. Therefore, the calculation of the numerical convective and conductive fluxes has a computational complexity of $\mathcal{O}(20n)$ per quadrature point, the same as $\mathcal{O}(20Rn)$ per face. Indeed, the calculation of the polynomial coefficients vectors is the most expensive part of the algorithm, with a computational complexity of $\mathcal{O}\left(6n^2\right)$ per face and cell, therefore growing quadratically with n.

5.4.2 Step 2

An optimization of step 2 is proposed to reduce the computational cost of determining the values and gradients of the polynomial reconstructions at the quadrature points on the faces and the collocation points.

5.4.2.1 Optimization

From the numerical scheme and the numerical treatment of the thermal coupling, the values of polynomial reconstruction $\widehat{\varphi}_i(\boldsymbol{x})$ and $\widehat{\varphi}_{iF}(\boldsymbol{x})$ and the gradients of polynomial reconstructions $\widetilde{\varphi}_{ij}(\boldsymbol{x})$ and $\widehat{\varphi}_{iF}(\boldsymbol{x})$ are determined, either at quadrature or collocation points, given as

$$\widehat{\varphi}_i(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_i \cdot \boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_i), \tag{5.34}$$

$$\widehat{\varphi}_{iF}(\boldsymbol{x}) = \widehat{\boldsymbol{\eta}}_{iF} \cdot \boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_{iF}), \tag{5.35}$$

$$\nabla\widetilde{\varphi}_{ij}(\boldsymbol{x}) = \left(\widetilde{\boldsymbol{\eta}}_{ij}^{\mathrm{T}} \nabla\boldsymbol{p}_d\left(\boldsymbol{x}-\boldsymbol{m}_{ij}\right)\right)^{\mathrm{T}}, \tag{5.36}$$

$$\nabla\widehat{\varphi}_{iF}(\boldsymbol{x}) = \left(\widehat{\boldsymbol{\eta}}_{iF}^{\mathrm{T}} \nabla\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_{iF})\right)^{\mathrm{T}}, \tag{5.37}$$

where $\nabla\boldsymbol{p}_d(\boldsymbol{x})$ is the Jacobian matrix associated to polynomial basis function $\boldsymbol{p}_d(\boldsymbol{x})$. Having the polynomial coefficient vectors given in Equations (5.31) and (5.33), then, Equations (5.34) and (5.37) are rewritten and rearranged such that vectors $\boldsymbol{b}_{ij}$, $\boldsymbol{b}_i$ and $\boldsymbol{b}_{iF}$ and boundary variable g_{iF} are put in evidence, given as

$$\begin{aligned}\widehat{\varphi}_i(\boldsymbol{x}) &= \left(\widehat{\boldsymbol{M}}_i\boldsymbol{b}_i\right)^{\mathrm{T}}\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_i) \\ &= \boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_i)\widehat{\boldsymbol{M}}_i\boldsymbol{b}_i,\end{aligned} \tag{5.38}$$

$$\begin{aligned}\widehat{\varphi}_{iF}(\boldsymbol{x}) &= \left(\widehat{\boldsymbol{M}}_{iF}\boldsymbol{b}_{iF} + \widehat{\boldsymbol{n}}_{iF}g_{iF}\right)^{\mathrm{T}}\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_{iF}) \\ &= \boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{M}}_{iF}\boldsymbol{b}_{iF} + g_{iF}\boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{n}}_{iF},\end{aligned} \tag{5.39}$$

$$\begin{aligned}\nabla\widetilde{\varphi}_{ij}(\boldsymbol{x}) &= \left(\left(\widetilde{\boldsymbol{M}}_{ij}\boldsymbol{b}_{ij}\right)^{\mathrm{T}}\nabla\boldsymbol{p}_d\left(\boldsymbol{x}-\boldsymbol{m}_{ij}\right)\right)^{\mathrm{T}} \\ &= \nabla\boldsymbol{p}_d^{\mathrm{T}}\left(\boldsymbol{x}-\boldsymbol{m}_{ij}\right)\widetilde{\boldsymbol{M}}_{ij}\boldsymbol{b}_{ij},\end{aligned} \tag{5.40}$$

$$\begin{aligned}\nabla\widehat{\varphi}_{iF}(\boldsymbol{x}) &= \left(\left(\widehat{\boldsymbol{M}}_{iF}\boldsymbol{b}_{iF} + g_{iF}\widehat{\boldsymbol{n}}_{iF}\right)^{\mathrm{T}}\nabla\boldsymbol{p}_d(\boldsymbol{x}-\boldsymbol{m}_{iF})\right)^{\mathrm{T}} \\ &= \nabla\boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{M}}_{iF}\boldsymbol{b}_{iF} + g_{iF}\nabla\boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{n}}_{iF},\end{aligned} \tag{5.41}$$

where vector $\boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x})$ and matrix $\nabla\boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x})$ are the transposes of vector $\boldsymbol{p}_d(\boldsymbol{x})$ and matrix $\nabla\boldsymbol{p}_d(\boldsymbol{x})$, respectively. For any point $\boldsymbol{x}$, function $\widehat{c}_{iF}(\boldsymbol{x})$, vector functions $\widehat{\boldsymbol{a}}_i(\boldsymbol{x})$, $\widehat{\boldsymbol{a}}_{iF}(\boldsymbol{x})$, $\widehat{\boldsymbol{c}}_{iF}(\boldsymbol{x})$, and matrix functions $\widetilde{\boldsymbol{A}}_{ij}(\boldsymbol{x})$ and $\widehat{\boldsymbol{A}}_{iF}(\boldsymbol{x})$, are defined respectively as

$$\widehat{c}_{iF}(\boldsymbol{x}) = \boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{n}}_{iF}, \tag{5.42}$$

$$\widehat{\boldsymbol{a}}_i(\boldsymbol{x}) = \boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_i)\widehat{\boldsymbol{M}}_i, \tag{5.43}$$

$$\widehat{\boldsymbol{a}}_{iF}(\boldsymbol{x}) = \boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{M}}_{iF}, \tag{5.44}$$

$$\widehat{\boldsymbol{c}}_{iF}(\boldsymbol{x}) = \nabla\boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{n}}_{iF}, \tag{5.45}$$

$$\widetilde{\boldsymbol{A}}_{ij}(\boldsymbol{x}) = \nabla\boldsymbol{p}_d^{\mathrm{T}}\left(\boldsymbol{x}-\boldsymbol{m}_{ij}\right)\widetilde{\boldsymbol{M}}_{ij}, \tag{5.46}$$

$$\widehat{\boldsymbol{A}}_{iF}(\boldsymbol{x}) = \nabla\boldsymbol{p}_d^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_{iF})\widehat{\boldsymbol{M}}_{iF}. \tag{5.47}$$

Then, Equations (5.38) and (5.41) are rewritten with the functions defined in Equations (5.42) to (5.47), given respectively as

$$\widehat{\varphi}_i(\boldsymbol{x}) = \widehat{\boldsymbol{a}}_i(\boldsymbol{x})\cdot\boldsymbol{b}_i, \tag{5.48}$$

$$\widehat{\varphi}_{iF}(\boldsymbol{x}) = \widehat{\boldsymbol{a}}_{iF}(\boldsymbol{x})\cdot\boldsymbol{b}_{iF} + g_{iF}\widehat{c}_{iF}(\boldsymbol{x}), \tag{5.49}$$

$$\nabla\widetilde{\varphi}_{ij}(\boldsymbol{x}) = \widetilde{\boldsymbol{A}}_{ij}(\boldsymbol{x})\boldsymbol{b}_{ij}, \tag{5.50}$$

$$\nabla\widehat{\varphi}_{iF}(\boldsymbol{x}) = \widehat{\boldsymbol{A}}_{iF}(\boldsymbol{x})\boldsymbol{b}_{iF} + g_{iF}\widehat{\boldsymbol{c}}_{iF}(\boldsymbol{x}). \tag{5.51}$$

5.4.2.2 Implementation

Vector functions $\widehat{\boldsymbol{a}}_i(\boldsymbol{x})$, $\widehat{\boldsymbol{a}}_{iF}(\boldsymbol{x})$, and $\widehat{c}_{iF}(\boldsymbol{x})$, matrix functions $\widetilde{\boldsymbol{A}}_{ij}(\boldsymbol{x})$ and $\widehat{\boldsymbol{A}}_{iF}(\boldsymbol{x})$, and function $\widehat{\boldsymbol{c}}_{iF}(\boldsymbol{x})$ only depend on geometrical parameters, namely the mesh, the polynomial degree, the linear constraint, stencil size and weights for the cost functional in the least-squares method, and point $\boldsymbol{x}$. Substituting point $\boldsymbol{x}$ with quadrature or collocation points, provide an expression that is computed

in advance at the pre-processing stage, stored in memory, and used in step 2 and, therefore, the optimization eliminates steps 1 from the implementation of the global residual operator. More specifically, the following vectors, matrices, and scalars are computed in advance at the pre-processing stage and stored in memory:

- For each inner face f_{ij} in computational subdomain Ω_{Δ}^{S}, numerical convective and conductive heat fluxes in Equations (5.18) and (5.19) use the values of polynomial reconstructions $\widehat{\varphi}_i(\boldsymbol{x})$ and $\widehat{\varphi}_j(\boldsymbol{x})$ and the gradient of polynomial reconstruction $\widetilde{\varphi}_{ij}(\boldsymbol{x})$ at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1, \ldots, R$, computed from Equations (5.48) and (5.50) having vectors $\widehat{\boldsymbol{a}}_i\left(\boldsymbol{q}_{ij,r}\right)$ and $\widehat{\boldsymbol{a}}_j\left(\boldsymbol{q}_{ij,r}\right)$ and matrix $\widetilde{\boldsymbol{A}}_{ij}\left(\boldsymbol{q}_{ij,r}\right)$ computed in advance.

- For each boundary face f_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$, numerical convective and conductive heat fluxes in Equations (5.20) and (5.21) use the values of polynomial reconstructions $\widehat{\varphi}_j(\boldsymbol{x})$ and $\widehat{\varphi}_{iF}(\boldsymbol{x})$ and the gradient of polynomial reconstruction $\widehat{\varphi}_{iF}(\boldsymbol{x})$ at quadrature points $\boldsymbol{q}_{Fj,r}$, $r = 1, \ldots, R$, computed from Equations (5.48), (5.49), and (5.51) having vectors $\widehat{\boldsymbol{a}}_i(\boldsymbol{q}_{iF,r})$, $\widehat{\boldsymbol{a}}_{iF}(\boldsymbol{q}_{iF,r})$, $\widehat{\boldsymbol{c}}_{iF}(\boldsymbol{q}_{iF,r})$, matrix $\widehat{\boldsymbol{A}}_{iF}(\boldsymbol{q}_{iF,r})$ and value $\widehat{c}_{iF}(\boldsymbol{q}_{iF,r})$ computed in advance.

- For each boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,A}}$, boundary variables in Equation (5.22) requires the value of polynomial reconstruction $\widehat{\varphi}_j(\boldsymbol{x})$, associated to counterpart boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$, at collocation point $\boldsymbol{p}_{i\mathrm{C}}$, computed from Equation (5.48) having vector $\widehat{\boldsymbol{a}}_{i\mathrm{C}}(\boldsymbol{p}_{i\mathrm{C}})$ computed in advance.

- For each boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$, boundary variables in Equation (5.23) requires the gradient of polynomial reconstruction $\widehat{\varphi}_{i\mathrm{C}}(\boldsymbol{x})$, associated to counterpart boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,A}}$, at collocation point $\boldsymbol{p}_{j\mathrm{C}}$, computed from Equation (5.51) having matrix $\widehat{\boldsymbol{A}}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)$ and vector $\widehat{\boldsymbol{c}}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)$ computed in advance.

The computational cost of the additional pre-processing stage is negligible, when compared with the cost of the iterative free-matrix method to solve the system of linear equations. On the other hand, the computational cost of the global residual operator is drastically reduced since the matrix-vector products in Equations (5.31) to (5.33), to compute the polynomial coefficients vectors of the polynomial reconstructions, in addition to the products in Equations (5.34) to (5.37), to compute the values and the gradients at the quadrature and collocation points, are substituted with simple dot products in Equations (5.48) to (5.51). Since the latter equations have a substantially lower computational cost, the overall performance of the proposed method is improved and, consequently, the execution time of the simulation is reduced.

5.4.2.3 Computational complexity

The computational complexity of the algorithm for the global residual operator, after optimization of both steps 1 and 2, is estimated in terms of n, assuming $s = 3n$ and taking only the highest polynomial term. The evaluation of the values and gradients of the polynomial reconstructions with Equations (5.48) to (5.51) is now the most expensive calculation of the algorithm, for which the corresponding number of floating-point operations and computational complexity are reported in Table 5.3.

Table 5.3: Floating-point operations and computational complexity for the calculation of the values and gradients of the polynomial reconstructions.

Equation	Product dimensions	Floating-point operations	Computational complexity
(5.48)	$s+1$	$2s+1$	$\mathcal{O}(6n)$
(5.49)	$s,\ 1$	$2s$	$\mathcal{O}(6n)$
(5.50)	$(3\times s)\times s$	$3(2s-1)$	$\mathcal{O}(18n)$
(5.51)	$(3\times s)\times s,\ 3$	$3(2s-1)+3$	$\mathcal{O}(18n)$

Having the values and gradients of the polynomial reconstructions in hand, the calculation of the numerical convective and conductive fluxes for the faces, followed by the residuals assembling for the cells, is inexpensive compared with the remaining algorithm. For the same reason, the evaluation of the polynomial reconstructions at the collocation points is not considered. Since each quadrature point on the faces requires one polynomial reconstruction value and gradient evaluations, then the algorithm has a computational complexity of $\mathcal{O}(24n)$ per quadrature point, the same as $\mathcal{O}(24Rn)$ per face. Compared with the optimization for step 1 only, the computational complexity of the algorithm grows linearly, instead of quadratically, with n.

Remark 22 The expected computational gains obtained from these optimizations cannot be deduced based on a direct comparison of the estimated computational complexity for the algorithm. Indeed, besides the floating-point operations, many other factors are at stake and have a rather difficult effect on the algorithm to take into account, such as cache management and compiler vectorization. Nevertheless, these predictions emphasize the importance of optimizing the algorithm to improve the computational efficiency of the proposed method.

5.4.3 Step 3

An optimization of step 3 is proposed to reduce the computational cost of evaluating the numerical heat fluxes on each inner and boundary face and the boundary variables at each collocation point.

5.4.3.1 Optimization

For each inner or boundary face f_{ij} in computational subdomain Ω_{Δ}^{S}, the associated mean-value of the numerical convective and conductive heat fluxes, denoted as $\mathcal{C}_{ij}$ and $\mathcal{D}_{ij}$, respectively, are given as

$$\mathcal{C}_{ij} = \sum_{r=1}^{R} \zeta_r \mathcal{C}_{ij,r}, \tag{5.52}$$

$$\mathcal{D}_{ij} = \sum_{r=1}^{R} \zeta_r \mathcal{D}_{ij,r}. \tag{5.53}$$

Affine expressions for the mean-values of the numerical heat fluxes as a function of the the cell mean-values are derived for the inner and boundary faces. Similarly is performed for the boundary variables.

5.4.3.1.1 Inner faces

For each inner face f_{ij} in computational subdomain Ω_{Δ}^{S}, Equations (5.52) and (5.53) are rewritten from Equations (5.18) and (5.19) as

$$\mathcal{C}_{ij} = \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij} \right]^{+} \widehat{\varphi}_i\left(\boldsymbol{q}_{ij,r}\right) + \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij} \right]^{-} \widehat{\varphi}_j\left(\boldsymbol{q}_{ij,r}\right), \tag{5.54}$$

$$\mathcal{D}_{ij} = \sum_{r=1}^{R} \zeta_r \left(-\kappa^{S}\left(\boldsymbol{q}_{ij,r}\right) \right) \nabla \widetilde{\varphi}_{ij}\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij}. \tag{5.55}$$

Polynomial reconstructions $\widehat{\varphi}_i(\boldsymbol{x})$ and $\widehat{\varphi}_j(\boldsymbol{x})$ evaluations at quadrature points $\boldsymbol{q}_{ij,r}$ in Equation (5.54) are determined with Equation (5.48), whereas the gradients of polynomial reconstruction $\widetilde{\varphi}_{ij}(\boldsymbol{x})$ at quadrature points $\boldsymbol{q}_{ij,r}$ in Equation (5.55) are determined with Equation (5.50). Therefore, the mean-value of the numerical convective and conductive heat fluxes are rewritten as

$$\begin{aligned} \mathcal{C}_{ij} = & \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij} \right]^{+} \left(\widehat{\boldsymbol{a}}_i\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{b}_i \right) \\ & + \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^{S}\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{s}_{ij} \right]^{-} \left(\widehat{\boldsymbol{a}}_j\left(\boldsymbol{q}_{ij,r}\right) \cdot \boldsymbol{b}_j \right), \end{aligned} \tag{5.56}$$

$$\mathcal{D}_{ij} = \sum_{r=1}^{R} \zeta_r \left(-\kappa^{S}\left(\boldsymbol{q}_{ij,r}\right) \right) \left(\left(\widetilde{\boldsymbol{A}}_{ij}\left(\boldsymbol{q}_{ij,r}\right) \right)^{\mathrm{T}} \boldsymbol{b}_{ij} \right) \cdot \boldsymbol{s}_{ij}. \tag{5.57}$$

Rearranging Equations (5.56) and (5.57), vectors $\boldsymbol{a}_{ij}^{\mathcal{C}+}, \boldsymbol{a}_{ij}^{\mathcal{C}-}, \boldsymbol{a}_{ij}^{\mathcal{D}} \in \mathbb{R}^s$ are defined respectively as

$$\boldsymbol{a}_{ij}^{\mathcal{C}+} = \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S \left(\boldsymbol{q}_{ij,r} \right) \cdot \boldsymbol{s}_{ij} \right]^+ \widehat{\boldsymbol{a}}_i \left(\boldsymbol{q}_{ij,r} \right), \tag{5.58}$$

$$\boldsymbol{a}_{ij}^{\mathcal{C}-} = \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S \left(\boldsymbol{q}_{ij,r} \right) \cdot \boldsymbol{s}_{ij} \right]^- \widehat{\boldsymbol{a}}_j \left(\boldsymbol{q}_{ij,r} \right), \tag{5.59}$$

$$\boldsymbol{a}_{ij}^{\mathcal{D}} = \sum_{r=1}^{R} \zeta_r \left(-\kappa^S \left(\boldsymbol{q}_{ij,r} \right) \right) \widetilde{\boldsymbol{A}}_{ij} \left(\boldsymbol{q}_{ij,r} \right) \boldsymbol{s}_{ij}, \tag{5.60}$$

such that vectors $\boldsymbol{b}_i$, $\boldsymbol{b}_j$ and $\boldsymbol{b}_{ij}$ are put in evidence. Then, Equations (5.56) and (5.57) are rewritten in an affine form with the cell mean-values, given as

$$\mathcal{C}_{ij} = \boldsymbol{a}_{ij}^{\mathcal{C}+} \cdot \boldsymbol{b}_i + \boldsymbol{a}_{ij}^{\mathcal{C}-} \cdot \boldsymbol{b}_j, \tag{5.61}$$

$$\mathcal{D}_{ij} = \boldsymbol{a}_{ij}^{\mathcal{D}} \cdot \boldsymbol{b}_{ij}. \tag{5.62}$$

5.4.3.1.2 Boundary faces

For each boundary face f_{iF} in computational boundary subset $\Gamma_{\Delta}^{F,S}$, Equations (5.52) and (5.53) are rewritten from Equations (5.20) and (5.21) as

$$\mathcal{C}_{iF} = \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S \left(\boldsymbol{q}_{iF,r} \right) \cdot \boldsymbol{s}_{iF} \right]^+ \widehat{\varphi}_i \left(\boldsymbol{q}_{iF,r} \right) + \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S \left(\boldsymbol{q}_{iF,r} \right) \cdot \boldsymbol{s}_{iF} \right]^- \widehat{\varphi}_{iF} \left(\boldsymbol{q}_{iF,r} \right), \tag{5.63}$$

$$\mathcal{D}_{iF} = \sum_{r=1}^{R} \zeta_r \left(-\kappa^S \left(\boldsymbol{q}_{iF,r} \right) \right) \nabla \widehat{\varphi}_{iF} \left(\boldsymbol{q}_{iF,r} \right) \cdot \boldsymbol{s}_{iF}, \tag{5.64}$$

Polynomial reconstructions $\widehat{\varphi}_i(\boldsymbol{x})$ and $\widehat{\varphi}_{iF}(\boldsymbol{x})$ evaluations at quadrature points $\boldsymbol{q}_{iF,r}$ in Equation (5.63) are determined with Equation (5.48) and (5.49), respectively, whereas the gradients of polynomial reconstruction $\widetilde{\varphi}_{iF}(\boldsymbol{x})$ at quadrature points $\boldsymbol{q}_{iF,r}$ in Equation (5.64) are determined with Equation (5.51). Therefore, the mean-value of the numerical convective and conductive heat fluxes are rewritten as

$$\begin{aligned} \mathcal{C}_{iF} = & \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S \left(\boldsymbol{q}_{iF,r} \right) \cdot \boldsymbol{s}_{iF} \right]^+ \left(\widehat{\boldsymbol{a}}_i \left(\boldsymbol{q}_{iF,r} \right) \cdot \boldsymbol{b}_i \right) \\ & + \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S \left(\boldsymbol{q}_{iF,r} \right) \cdot \boldsymbol{s}_{iF} \right]^- \left(\widehat{\boldsymbol{a}}_{iF} \left(\boldsymbol{q}_{iF,r} \right) \cdot \boldsymbol{b}_{iF} + g_{iF} \widehat{c}_{iF} \left(\boldsymbol{q}_{iF,r} \right) \right), \end{aligned} \tag{5.65}$$

$$\mathcal{D}_{iF} = \sum_{r=1}^{R} \zeta_r \left(-\kappa^S \left(\boldsymbol{q}_{iF,r} \right) \right) \left(\left(\widehat{\boldsymbol{A}}_{iF} \left(\boldsymbol{q}_{iF,r} \right) \right)^{\mathrm{T}} \boldsymbol{b}_{iF} + g_{iF} \widehat{\boldsymbol{c}}_{iF} \left(\boldsymbol{q}_{iF,r} \right) \right) \cdot \boldsymbol{s}_{iF}. \tag{5.66}$$

Rearranging Equations (5.65) and (5.66), vectors $\boldsymbol{a}_{iF}^{\mathcal{C}+}, \boldsymbol{a}_{iF}^{\mathcal{C}-}, \boldsymbol{a}_{iF}^{\mathcal{D}} \in \mathbb{R}^s$ and values $c_{iF}^{\mathcal{C}-}, c_{iF}^{\mathcal{D}} \in \mathbb{R}$ are defined respectively as

$$\boldsymbol{a}_{iF}^{\mathcal{C}+} = \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S\left(\boldsymbol{q}_{iF,r}\right) \cdot \boldsymbol{s}_{iF}\right]^{+} \widehat{\boldsymbol{a}}_i\left(\boldsymbol{q}_{iF,r}\right), \tag{5.67}$$

$$\boldsymbol{a}_{iF}^{\mathcal{C}-} = \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S\left(\boldsymbol{q}_{iF,r}\right) \cdot \boldsymbol{s}_{iF}\right]^{-} \widehat{\boldsymbol{a}}_{iF}\left(\boldsymbol{q}_{iF,r}\right), \tag{5.68}$$

$$\boldsymbol{a}_{iF}^{\mathcal{D}} = \sum_{r=1}^{R} \zeta_r \left(-\kappa^S\left(\boldsymbol{q}_{iF,r}\right)\right) \widetilde{\boldsymbol{A}}_{iF}\left(\boldsymbol{q}_{iF,r}\right) \boldsymbol{s}_{iF}, \tag{5.69}$$

$$c_{iF}^{\mathcal{C}-} = \sum_{r=1}^{R} \zeta_r \left[\boldsymbol{u}^S\left(\boldsymbol{q}_{iF,r}\right) \cdot \boldsymbol{s}_{iF}\right]^{-} \widehat{c}_{iF}\left(\boldsymbol{q}_{iF,r}\right), \tag{5.70}$$

$$c_{iF}^{\mathcal{D}} = \sum_{r=1}^{R} \zeta_r \left(-\kappa^S\left(\boldsymbol{q}_{iF,r}\right)\right) \widehat{\boldsymbol{c}}_{iF}\left(\boldsymbol{q}_{iF,r}\right) \cdot \boldsymbol{s}_{iF}, \tag{5.71}$$

such that vectors $\boldsymbol{b}_i$ and $\boldsymbol{b}_{iF}$ and boundary variable g_{iF} are put in evidence. Then, Equations (5.65) and (5.66) are rewritten in an affine form with the cell mean-values and boundary variables, given as

$$\mathcal{C}_{iF} = \boldsymbol{a}_{iF}^{\mathcal{C}+} \cdot \boldsymbol{b}_i + \boldsymbol{a}_{iF}^{\mathcal{C}-} \cdot \boldsymbol{b}_{iF} + g_{iF} c_{iF}^{\mathcal{C}-}, \tag{5.72}$$

$$\mathcal{D}_{iF} = \boldsymbol{a}_{iF}^{\mathcal{D}} \cdot \boldsymbol{b}_{iF} + g_{iF} c_{iF}^{\mathcal{D}}. \tag{5.73}$$

5.4.3.1.3 Boundary variables

For each boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,A}}$ and counterpart boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$, boundary variables $g_{i\mathrm{C}}$ and $g_{j\mathrm{C}}$ at collocation points $\boldsymbol{p}_{i\mathrm{C}}$ and $\boldsymbol{p}_{j\mathrm{C}}$, respectively, are given in Equations (5.22) and (5.23), respectively. Polynomial reconstructions $\widehat{\varphi}_j(\boldsymbol{x})$ evaluation at collocation point $\boldsymbol{p}_{i\mathrm{C}}$ is determined with Equation (5.48), whereas the gradient of polynomial reconstruction $\widehat{\varphi}_{i\mathrm{C}}(\boldsymbol{x})$ at collocation point $\boldsymbol{p}_{j\mathrm{C}}$ is determined with Equation (5.51). Therefore, Equations (5.22) and (5.23) are rewritten as

$$g_{i\mathrm{C}} = \widehat{\boldsymbol{a}}_j\left(\boldsymbol{p}_{i\mathrm{C}}\right) \cdot \boldsymbol{b}_j, \tag{5.74}$$

$$g_{j\mathrm{C}} = -\kappa^{\mathrm{A}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\left(\left(\widehat{\boldsymbol{A}}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\right)^{\mathrm{T}} \boldsymbol{b}_{i\mathrm{C}} + g_{i\mathrm{C}} \widehat{\boldsymbol{c}}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\right) \cdot \boldsymbol{v}_{j\mathrm{C}}. \tag{5.75}$$

Rearranging Equations (5.74) and (5.75), vectors $\boldsymbol{a}_{i\mathrm{C}}^{\mathcal{B}}, \boldsymbol{a}_{j\mathrm{C}}^{\mathcal{B}} \in \mathbb{R}^s$ and value $c_{i\mathrm{C}}^{\mathcal{B}} \in \mathbb{R}$ are defined respectively as

$$\boldsymbol{a}_{i\mathrm{C}}^{\mathcal{B}} = \widehat{\boldsymbol{a}}_j\left(\boldsymbol{p}_{i\mathrm{C}}\right), \tag{5.76}$$

$$\boldsymbol{a}_{j\mathrm{C}}^{\mathcal{B}} = -\kappa^{S}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\widehat{\boldsymbol{A}}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\boldsymbol{v}_{j\mathrm{C}}, \tag{5.77}$$

$$c_{j\mathrm{C}}^{\mathcal{B}} = -\kappa^{S}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\widehat{\boldsymbol{c}}_{i\mathrm{C}}\left(\boldsymbol{p}_{j\mathrm{C}}\right)\cdot\boldsymbol{v}_{j\mathrm{C}}, \tag{5.78}$$

such that vectors $\boldsymbol{b}_j$ and $\boldsymbol{b}_{i\mathrm{C}}$ and boundary variable $g_{i\mathrm{C}}$ are put in evidence. Then, Equations (5.74) and (5.75) are rewritten in an affine form with the cell mean-values, given as

$$g_{i\mathrm{C}} = \boldsymbol{a}_{i\mathrm{C}}^{\mathcal{B}}\cdot\boldsymbol{b}_j, \tag{5.79}$$

$$g_{j\mathrm{C}} = \boldsymbol{a}_{j\mathrm{C}}^{\mathcal{B}}\cdot\boldsymbol{b}_{i\mathrm{C}} + g_{i\mathrm{C}}c_{j\mathrm{C}}^{\mathcal{B}}. \tag{5.80}$$

5.4.3.2 Implementation

Vectors $\boldsymbol{a}_{ij}^{\mathcal{C}+}$, $\boldsymbol{a}_{ij}^{\mathcal{C}-}$, $\boldsymbol{a}_{ij}^{\mathcal{D}}$, $\boldsymbol{a}_{iF}^{\mathcal{C}+}$, $\boldsymbol{a}_{iF,j}^{\mathcal{C}}$, $\boldsymbol{a}_{iF}^{\mathcal{D}}$, $\boldsymbol{a}_{i\mathrm{C}}^{\mathcal{B}}$, and $\boldsymbol{a}_{j\mathrm{C}}^{\mathcal{B}}$ together with scalars $c_{iF}^{\mathcal{C}-}$, $c_{iF}^{\mathcal{D}}$, and $c_{i\mathrm{C}}^{\mathcal{B}}$ only depend on geometrical parameters, namely the mesh, the polynomial degree, the linear constraint, stencil size and weights for the cost functional in the least-squares method, and quadrature or collocation points. On the other side, vectors $\boldsymbol{b}_{ij}$, $\boldsymbol{b}_i$, and $\boldsymbol{b}_{iF}$ gather the cell mean-values (internal variables) associated to the cells with an index in stencils $\mathcal{S}_{ij}$, $\mathcal{S}_i$, and $\mathcal{S}_{iF}$, respectively, and g_{iF} is a boundary variable. Therefore, vectors $\boldsymbol{a}_{ij}^{\mathcal{C}+}$, $\boldsymbol{a}_{ij}^{\mathcal{C}-}$, $\boldsymbol{a}_{ij}^{\mathcal{D}}$, $\boldsymbol{a}_{iF}^{\mathcal{C}+}$, $\boldsymbol{a}_{iF,j}^{\mathcal{C}}$, $\boldsymbol{a}_{iF}^{\mathcal{D}}$, $\boldsymbol{a}_{i\mathrm{C}}^{\mathcal{B}}$, and $\boldsymbol{a}_{j\mathrm{C}}^{\mathcal{B}}$ together with scalars $c_{iF}^{\mathcal{C}-}$, $c_{iF}^{\mathcal{D}}$, and $c_{i\mathrm{C}}^{\mathcal{B}}$ are computed in advance at the pre-processing stage, stored in memory, and used in step 4 and, therefore, the optimization eliminates steps 1 and 2 from the implementation of the global residual operator.

The computational cost of the additional pre-processing stage is negligible, when compared with the cost of the iterative free-matrix method to solve the system of linear equations. On the other hand, the computational cost of the global residual operator is drastically reduced since it replaces the dot products in Equations (5.54), (5.55), (5.63), and (5.64) with only one dot product in Equations (5.61), (5.62), (5.72), and (5.73), respectively. Since the latter equations have a lower computational cost, with a factor of around R, the overall performance of the proposed method is improved and, consequently, the execution time is reduced.

5.4.3.3 Computational complexity

The computational complexity of the algorithm for the global residual operator is straightforward to estimate after optimization of steps 1 to 3. Indeed, the calculation of the numerical convective and conductive fluxes requires, roughly speaking, a single dot product of dimension s or $s+1$ for each. Notice that, the residuals assembling for the cells and the calculation of the boundary variables are inexpensive compared with the remaining algorithm, and, for such reason, they are not taken into account. Therefore, assuming $s = 3n$ and taking only the highest polynomial term, the complexity of the algorithm is $\mathcal{O}(12n)$ per face and grows linearly with n.

5.4.4 Step 4

An optimization of step 4 is proposed to reduce the computational cost of computing the residual for the cells. Similarly, the relation of the thermal coupling for the boundary variables is also calculated in an improved way. An explicit formulation of the proposed method with a system of linear equations in terms of the cells mean-values and boundary variables is provided replacing the global residual operator.

5.4.4.1 Local and global mappings

For each mesh element, $\mathcal{S}$ is a stencil gathering the global indices of s cells in the vicinity of the reference point, with $\mathcal{S} \subset \{1, \ldots, n^S\}$. For the least-squares method, local index $\ell \in \{1, \ldots, s\}$ is a fixed and unique position specified beforehand for each global index $k \in \mathcal{S}$, which ascertains the associated rows in the coefficients matrix derived from the cost functional and the associated positions in the vector of the cell mean-values. A global index mapping is defined as the correspondence from local index ℓ to global index k, denoted as $\ell \xrightarrow{\mathrm{G}} k$, whereas a local index mapping is defined as the opposite operation, denoted as $k \xrightarrow{\mathrm{L}} \ell$.

A global vector mapping consists in transforming any local vector $\boldsymbol{a} \in \mathbb{R}^s$ into global vector $\boldsymbol{b} \in \mathbb{R}^{n^S}$, having entries $\boldsymbol{b}[k] = \boldsymbol{a}[\ell]$ with index mapping from local to global indices $\ell \xrightarrow{\mathrm{G}} k$ and the remaining entries are zeros. The global vector mapping operation is performed with matrix $\boldsymbol{S} \in \mathbb{R}^{n^S \times s}$ defined as a binary matrix with the entries $\boldsymbol{S}[k, \ell] = 1$ with index mapping from local to global indices $\ell \xrightarrow{\mathrm{G}} k$ and the remaining entries are zeros. Therefore, all columns are non-null having a single non-null entry, whereas only the rows corresponding to global indices k are non-null (matrix $\boldsymbol{S}$ is full column rank by construction). In turn, a local vector mapping consists in transforming any global vector $\boldsymbol{c} \in \mathbb{R}^{n^S}$ into local vector $\boldsymbol{a} \in \mathbb{R}^s$, having entries $\boldsymbol{a}[\ell] = \boldsymbol{c}[k]$ with index mapping from global to local indices $k \xrightarrow{\mathrm{L}} \ell$ (there are no remaining entries in that case). The local vector mapping operation is performed with matrix $\boldsymbol{S}^{\mathrm{T}} \in \mathbb{R}^{s \times n^S}$ defined as the transpose of matrix $\boldsymbol{S}$ for the global vector mapping (matrix $\boldsymbol{S}^{\mathrm{T}}$ is full row rank by construction).

Remark 23 For instance, consider a mesh with 6 cells and the associated cell mean-values are gathered in vector $\boldsymbol{\Phi}^S = (a, b, c, d, e, f)$. Consider that a mesh element has stencil $\mathcal{S} = \{1, 4, 5\}$ and, therefore, the vector from the least-squares method gathering the associated cell mean-values is

$\boldsymbol{a} = (a, d, e)$. The global vector mapping operation is performed with matrix $\boldsymbol{S}$ given as

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.81}$$

The global vector mapping applied to vector $\boldsymbol{a}$ and the local vector mapping applied to vector $\boldsymbol{\Phi}^S$, which recovers vector $\boldsymbol{a}$, are given respectively as

$$\boldsymbol{Sa} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ d \\ e \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \\ d \\ e \\ 0 \end{bmatrix}, \tag{5.82}$$

$$\boldsymbol{S}^T\boldsymbol{\Phi}^S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{5.83}$$

Property 1 Any two vectors $\boldsymbol{a}, \boldsymbol{d} \in \mathbb{R}^s$ satisfy the property $\boldsymbol{a} \cdot \boldsymbol{d} = [\boldsymbol{Sa}] \cdot [\boldsymbol{Sd}]$ for matrix $\boldsymbol{S} \in \mathbb{R}^{n^S \times s}$ associated with a global vector mapping. Indeed, the entries in vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ are mapped to the same positions in vectors $\boldsymbol{Sa}$ and $\boldsymbol{Sc}$, whereas the remaining entries are .

Property 2 Any two vectors $\boldsymbol{a}, \boldsymbol{d} \in \mathbb{R}^s$ and vector $\boldsymbol{\Phi}^S$ satisfy the property $[\boldsymbol{Sa}] \cdot [\boldsymbol{Sd}] = [\boldsymbol{Sa}] \cdot \boldsymbol{\Phi}^S$ provided that $\boldsymbol{d} = \boldsymbol{S}^T\boldsymbol{\Phi}^S$ for matrix $\boldsymbol{S} \in \mathbb{R}^{n^S \times s}$ associated with a global vector mapping. Indeed, $[\boldsymbol{Sa}] \cdot [\boldsymbol{Sd}] = \boldsymbol{a}^T\boldsymbol{S}^T\boldsymbol{Sd} = \boldsymbol{a}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{\Phi}^S$ and the full rank implies that $\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{S}^T$ since a local vector mapping applied after a global vector mapping does not lose information (the original vector is obtained if applied matrix $\boldsymbol{S}^T\boldsymbol{S}$). Therefore, $\boldsymbol{a}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{\Phi}^S = \boldsymbol{a}^T\boldsymbol{S}^T\boldsymbol{\Phi}^S = [\boldsymbol{Sa}] \cdot \boldsymbol{\Phi}^S$ and the property $[\boldsymbol{Sa}] \cdot [\boldsymbol{Sd}] = [\boldsymbol{Sa}] \cdot \boldsymbol{\Phi}^S$ holds.

For each cell c_i in computational subdomain Ω_Δ^S, for each inner face f_{ij} in computational subdomain Ω_Δ^S, and for each boundary face f_{iF} on computational boundary subset $\Gamma_\Delta^{F,S}$, the matrices associated to the global vector mappings are defined according to associated stencils $\mathcal{S}_i$, $\mathcal{S}_{ij}$, and $\mathcal{S}_{iF}$, respectively, and are denoted as $\boldsymbol{S}_i$, $\boldsymbol{S}_{ij}$, and $\boldsymbol{S}_{iF} \in \mathbb{R}^{n^S \times s}$, respectively.

Vector $\boldsymbol{\Psi}^{F,S}$ gathers the boundary variables for each boundary face f_{iF} on computational boundary subset $\Gamma_\Delta^{F,S}$, such that there is an index $\ell \in \{1, \ldots, n^{F,S}\}$ providing $\boldsymbol{\Psi}^{F,S}[\ell] = g_{iF}$. In that regard, vector $\boldsymbol{e}_{iF} \in \mathbb{R}^{n^{F,S}}$ is defined having a single non-null entry $\boldsymbol{e}_{iF}[\ell] = 1$ and the remaining entries are zeros, such that $g_{iF} = \boldsymbol{e}_{iF} \cdot \boldsymbol{\Psi}^{F,S}$.

5.4.4.2 Optimization

From Properties 1 and 2, the mean-value of the numerical heat fluxes and boundary variables are rewritten in a global vector mapping representation, as follows:

- For each inner face f_{ij} in computational subdomain Ω_Δ^S, Equations (5.61) and (5.62) are rewritten as

$$\mathcal{C}_{ij} = \left(\boldsymbol{S}_i \boldsymbol{a}_{ij,i}^{\mathcal{C}}\right) \cdot \left(\boldsymbol{S}_i \boldsymbol{b}_i\right) + \left(\boldsymbol{S}_j \boldsymbol{a}_{ij,j}^{\mathcal{C}}\right) \cdot \left(\boldsymbol{S}_j \boldsymbol{b}_j\right), \tag{5.84}$$

$$\mathcal{D}_{ij} = \left(\boldsymbol{S}_{ij} \boldsymbol{a}_{ij}^{\mathcal{D}}\right) \cdot \left(\boldsymbol{S}_{ij} \boldsymbol{b}_{ij}\right). \tag{5.85}$$

  Since $\boldsymbol{b}_i = \boldsymbol{S}_i^{\mathrm{T}} \boldsymbol{\Phi}^S$, $\boldsymbol{b}_j = \boldsymbol{S}_j^{\mathrm{T}} \boldsymbol{\Phi}^S$, and $\boldsymbol{b}_{ij} = \boldsymbol{S}_{ij}^{\mathrm{T}} \boldsymbol{\Phi}^S$, Equations (5.84) and (5.85) are rewritten as

$$\mathcal{C}_{ij} = \left(\boldsymbol{S}_i \boldsymbol{a}_{ij,i}^{\mathcal{C}} + \boldsymbol{S}_j \boldsymbol{a}_{ij,j}^{\mathcal{C}}\right) \cdot \boldsymbol{\Phi}^S, \tag{5.86}$$

$$\mathcal{D}_{ij} = \left(\boldsymbol{S}_{ij} \boldsymbol{a}_{ij}^{\mathcal{D}}\right) \cdot \boldsymbol{\Phi}^S. \tag{5.87}$$

- For each boundary face f_{iF} on computational boundary subset $\Gamma_\Delta^{F,S}$, Equations (5.72) and (5.73) are rewritten as

$$\mathcal{C}_{iF} = \left(\boldsymbol{S}_i \boldsymbol{a}_{iF,i}^{\mathcal{C}}\right) \cdot \left(\boldsymbol{S}_i \boldsymbol{b}_i\right) + \left(\boldsymbol{S}_{iF} \boldsymbol{a}_{iF,iF}^{\mathcal{C}}\right) \cdot \left(\boldsymbol{S}_{iF} \boldsymbol{b}_{iF}\right) + \left(\boldsymbol{e}_{iF} g_{iF}\right) \cdot \left(\boldsymbol{e}_{iF} c_{iF}^{\mathcal{C}}\right), \tag{5.88}$$

$$\mathcal{D}_{iF} = \left(\boldsymbol{S}_{iF} \boldsymbol{a}_{iF}^{\mathcal{D}}\right) \cdot \left(\boldsymbol{S}_{iF} \boldsymbol{b}_{iF}\right) + \left(\boldsymbol{e}_{iF} c_{iF}^{\mathcal{D}}\right) \cdot \left(\boldsymbol{e}_{iF} g_{iF}\right). \tag{5.89}$$

  Since $\boldsymbol{b}_i = \boldsymbol{S}_i^{\mathrm{T}} \boldsymbol{\Phi}^S$, $\boldsymbol{b}_{iF} = \boldsymbol{S}_{iF}^{\mathrm{T}} \boldsymbol{\Phi}^S$, and $g_{iF} = \boldsymbol{e}_{iF}^{\mathrm{T}} \boldsymbol{\Psi}^{F,S}$, Equations (5.88) and (5.89) are rewritten as

$$\mathcal{C}_{iF} = \left(\boldsymbol{S}_i \boldsymbol{a}_{iF,i}^{\mathcal{C}} + \boldsymbol{S}_{iF} \boldsymbol{a}_{iF,iF}^{\mathcal{C}}\right) \cdot \boldsymbol{\Phi}^S + \left(\boldsymbol{e}_{iF} c_{iF}^{\mathcal{C}}\right) \cdot \boldsymbol{\Psi}^{F,S}, \tag{5.90}$$

$$\mathcal{D}_{iF} = \left(\boldsymbol{S}_{iF} \boldsymbol{a}_{iF}^{\mathcal{D}}\right) \cdot \boldsymbol{\Phi}^S + \left(\boldsymbol{e}_{iF} c_{iF}^{\mathcal{D}}\right) \cdot \boldsymbol{\Psi}^{F,S}. \tag{5.91}$$

- For each boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,A}}$ and counterpart boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_{\Delta}^{\mathrm{C,B}}$, Equations (5.79) and (5.80) are rewritten as

$$g_{i\mathrm{C}} = \left(\boldsymbol{S}_j \boldsymbol{a}_{i\mathrm{C}}^{\mathcal{B}}\right) \cdot \left(\boldsymbol{S}_j \boldsymbol{b}_j\right), \tag{5.92}$$

$$g_{j\mathrm{C}} = \left(\boldsymbol{S}_{i\mathrm{C}} \boldsymbol{a}_{j\mathrm{C}}^{\mathcal{B}}\right) \cdot \left(\boldsymbol{S}_{i\mathrm{C}} \boldsymbol{b}_{i\mathrm{C}}\right) + \left(\boldsymbol{e}_{i\mathrm{C}} g_{i\mathrm{C}}\right) \cdot \left(\boldsymbol{e}_{i\mathrm{C}} c_{j\mathrm{C}}^{\mathcal{B}}\right). \tag{5.93}$$

  Since $\boldsymbol{b}_j = \boldsymbol{S}_j^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{B}}$, $\boldsymbol{b}_{i\mathrm{C}} = \boldsymbol{S}_{i\mathrm{C}}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{A}}$, and $g_{i\mathrm{C}} = \boldsymbol{e}_{i\mathrm{C}}^{\mathrm{T}} \boldsymbol{\Psi}^{\mathrm{C,A}}$, Equations (5.92) and (5.93) are rewritten as

$$g_{i\mathrm{C}} = \left(\boldsymbol{S}_j \boldsymbol{a}_{i\mathrm{C}}^{\mathcal{B}}\right) \cdot \boldsymbol{\Phi}^{\mathrm{B}}, \tag{5.94}$$

$$g_{j\mathrm{C}} = \left(\boldsymbol{S}_{i\mathrm{C}} \boldsymbol{a}_{j\mathrm{C}}^{\mathcal{B}}\right) \cdot \boldsymbol{\Phi}^{\mathrm{A}} + \left(\boldsymbol{e}_{i\mathrm{C}} c_{j\mathrm{C}}^{\mathcal{B}}\right) \cdot \boldsymbol{\Psi}^{\mathrm{C,A}}. \tag{5.95}$$

Then, the equations derived from the discretization of the subproblems and the numerical treatment of the thermal coupling are rewritten in terms of affine residual operators, as follows:

- For each cell c_i in computational subdomain Ω_{Δ}^{S} with only inner faces, the residual given in Equation (5.24) is rewritten from Equations (5.86) and (5.87) as

$$\mathcal{R}_i = \sum_{j \in \mathcal{N}_i} \left|f_{ij}\right| \left(\left(\boldsymbol{S}_i \boldsymbol{a}_{ij,i}^{\mathcal{C}} + \boldsymbol{S}_j \boldsymbol{a}_{ij,j}^{\mathcal{C}}\right) \cdot \boldsymbol{\Phi}^{S} + \left(\boldsymbol{S}_{ij} \boldsymbol{a}_{ij}^{\mathcal{D}}\right) \cdot \boldsymbol{\Phi}^{S} \right) - f_i |c_i|, \tag{5.96}$$

  Defining vector $\boldsymbol{r}_i \in \mathbb{R}^{n^S}$ as

$$\boldsymbol{r}_i = \sum_{j \in \mathcal{N}_i} \left|f_{ij}\right| \left(\boldsymbol{S}_i \boldsymbol{a}_{ij,i}^{\mathcal{C}} + \boldsymbol{S}_j \boldsymbol{a}_{ij,j}^{\mathcal{C}} + \boldsymbol{S}_{ij} \boldsymbol{a}_{ij}^{\mathcal{D}} \right), \tag{5.97}$$

  then, Equation (5.96) is rewritten as

$$\mathcal{R}_i = \boldsymbol{r}_i \cdot \boldsymbol{\Phi}^{S} - f_i |c_i|. \tag{5.98}$$

- For each cell c_i in computational subdomain Ω_{Δ}^{S} with a boundary face f_{iF} on computational boundary subset $\Gamma_{\Delta}^{F,S}$, the residual given in Equation (5.24) is rewritten from Equations (5.90) and (5.91) as

$$\begin{aligned} \mathcal{R}_i = & \sum_{j \in \mathcal{N}_i \setminus F} \left|f_{ij}\right| \left(\left(\boldsymbol{S}_i \boldsymbol{a}_{ij}^{\mathcal{C}} + \boldsymbol{S}_j \boldsymbol{a}_{ij,j}^{\mathcal{C}}\right) \cdot \boldsymbol{\Phi}^{S} + \left(\boldsymbol{S}_{ij} \boldsymbol{a}_{ij}^{\mathcal{D}}\right) \cdot \boldsymbol{\Phi}^{S} \right) \\ & + \left|f_{iF}\right| \left(\left(\boldsymbol{S}_i \boldsymbol{a}_{iF,i}^{\mathcal{C}} + \boldsymbol{S}_{iF} \boldsymbol{a}_{iF,iF}^{\mathcal{C}}\right) \cdot \boldsymbol{\Phi}^{S} + \left(\boldsymbol{e}_{iF} c_{iF}^{\mathcal{C}}\right) \cdot \boldsymbol{\Psi}^{F,S} \right. \\ & \left. + \left(\boldsymbol{S}_{iF} \boldsymbol{a}_{iF}^{\mathcal{D}}\right) \cdot \boldsymbol{\Phi}^{S} + \left(\boldsymbol{e}_{iF} c_{iF}^{\mathcal{D}}\right) \cdot \boldsymbol{\Psi}^{F,S} \right) - f_i |c_i|. \end{aligned} \tag{5.99}$$

Defining vectors $\boldsymbol{r}_i, \boldsymbol{t}_i \in \mathbb{R}^{n^S}$ as

$$\begin{aligned}\boldsymbol{r}_i = &\sum_{j \in \mathcal{N}_i \setminus F} |f_{ij}| \left(\boldsymbol{S}_i \boldsymbol{a}^{\mathcal{C}}_{ij,i} + \boldsymbol{S}_j \boldsymbol{a}^{\mathcal{C}}_{ij,j} + \boldsymbol{S}_{ij} \boldsymbol{a}^{\mathcal{D}}_{ij} \right) \\ &+ |f_{iF}| \left(\boldsymbol{S}_i \boldsymbol{a}^{\mathcal{C}}_{iF,i} + \boldsymbol{S}_{iF} \boldsymbol{a}^{\mathcal{C}}_{iF,iF} + \boldsymbol{S}_{iF} \boldsymbol{a}^{\mathcal{D}}_{iF} \right),\end{aligned} \tag{5.100}$$

$$\boldsymbol{t}_i = |f_{iF}| \left(\boldsymbol{e}_{iF} c^{\mathcal{C}}_{iF} + \boldsymbol{e}_{iF} c^{\mathcal{D}}_{iF} \right), \tag{5.101}$$

then, Equation (5.99) is rewritten as

$$\mathcal{R}_i = \boldsymbol{r}_i \cdot \boldsymbol{\Phi}^S + \boldsymbol{t}_i \cdot \boldsymbol{\Psi}^{F,S} - f_i |c_i|. \tag{5.102}$$

- For each boundary face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma^{\mathrm{C,A}}_{\Delta}$ and counterpart boundary face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma^{\mathrm{C,B}}_{\Delta}$, the residuals given in Equations (5.25) and (5.26), respectively, are rewritten from Equations (5.94) and (5.95) as

$$\mathcal{G}_{i\mathrm{C}} = \left(\boldsymbol{S}_j \boldsymbol{a}^{\mathcal{B}}_{i\mathrm{C}} \right) \cdot \boldsymbol{\Phi}^{\mathrm{B}} - g_{i\mathrm{C}}, \tag{5.103}$$

$$\mathcal{G}_{j\mathrm{C}} = \left(\boldsymbol{S}_{i\mathrm{C}} \boldsymbol{a}^{\mathcal{B}}_{j\mathrm{C}} \right) \cdot \boldsymbol{\Phi}^{\mathrm{A}} + \left(\boldsymbol{e}_{i\mathrm{C}} c^{\mathcal{B}}_{j\mathrm{C}} \right) \cdot \boldsymbol{\Psi}^{\mathrm{C,A}} - g_{j\mathrm{C}}. \tag{5.104}$$

Notice that $g_{i\mathrm{C}} = \boldsymbol{e}_{i\mathrm{C}} \cdot \boldsymbol{\Psi}^{\mathrm{C,A}}$ and $g_{j\mathrm{C}} = \boldsymbol{e}_{j\mathrm{C}} \cdot \boldsymbol{\Psi}^{\mathrm{C,B}}$. Defining vectors $\boldsymbol{q}_{i\mathrm{C}} \in \mathbb{R}^{n^{\mathrm{B}}}$ and $\boldsymbol{q}_{j\mathrm{C}} \in \mathbb{R}^{n^{\mathrm{A}}}$ as

$$\boldsymbol{q}_{i\mathrm{C}} = \boldsymbol{S}_j \boldsymbol{a}^{\mathcal{B}}_{i\mathrm{C}}, \tag{5.105}$$

$$\boldsymbol{q}_{j\mathrm{C}} = \boldsymbol{S}_{i\mathrm{C}} \boldsymbol{a}^{\mathcal{B}}_{j\mathrm{C}}, \tag{5.106}$$

$$\boldsymbol{w}_{j\mathrm{C}} = \boldsymbol{e}_{i\mathrm{C}} c^{\mathcal{B}}_{j\mathrm{C}}, \tag{5.107}$$

then, Equations (5.103) and (5.104) are rewritten as

$$\mathcal{G}_{i\mathrm{C}} = \boldsymbol{q}_{i\mathrm{C}} \cdot \boldsymbol{\Phi}^{\mathrm{B}} - \boldsymbol{e}_{i\mathrm{C}} \cdot \boldsymbol{\Psi}^{\mathrm{C,A}}, \tag{5.108}$$

$$\mathcal{G}_{j\mathrm{C}} = \boldsymbol{q}_{j\mathrm{C}} \cdot \boldsymbol{\Phi}^{\mathrm{A}} + \boldsymbol{w}_{j\mathrm{C}} \cdot \boldsymbol{\Psi}^{\mathrm{C,A}} - \boldsymbol{e}_{j\mathrm{C}} \cdot \boldsymbol{\Psi}^{\mathrm{C,B}}. \tag{5.109}$$

5.4.4.3 Explicit formulation

Having the residuals of the implicit formulation of the proposed method given as dot products with vectors $\boldsymbol{\Phi}^S$ and vectors $\boldsymbol{\Psi}^{F,S}$ in evidence, then, the associated residual operators can be assembled in matrix form, as follows:

- The residuals defined in Equation (5.98) for cells c_i in computational subdomain Ω^S_{Δ} with only inner faces are rewritten in vector form with matrix $\boldsymbol{R}^S \in \mathbb{R}^{(n^S - n^{\mathrm{R},S} - n^{\mathrm{C},S}) \times n^S}$ composed of rows

corresponding to vectors $\boldsymbol{r}_i$ and vector $\boldsymbol{f}^S \in \mathbb{R}^{n^S - n^{\mathrm{R},S} - n^{\mathrm{C},S}}$ with entries $f_i|c_i|$, given as

$$\mathcal{R}^S\left(\boldsymbol{\Phi}^S\right) = \boldsymbol{R}^S\boldsymbol{\Phi}^S - \boldsymbol{f}^S. \tag{5.110}$$

- The residuals defined in Equation (5.102) for cells c_i in computational subdomain Ω_Δ^S with a boundary face f_{iF} on computational boundary subset $\Gamma_\Delta^{F,S}$ are rewritten in vector form with matrix $\boldsymbol{R}^{F,S} \in \mathbb{R}^{n^{F,S} \times n^S}$ composed of rows corresponding to vectors $\boldsymbol{r}_i$, matrix $\boldsymbol{T}^{F,S} \in \mathbb{R}^{n^{F,S} \times n^{F,S}}$ composed of rows corresponding to vectors $\boldsymbol{t}_i$, and vector $\boldsymbol{f}^S \in \mathbb{R}^{n^S}$ with entries $f_i|c_i|$, given as

$$\mathcal{R}^{F,S}\left(\boldsymbol{\Phi}^S, \boldsymbol{\Psi}^{F,S}\right) = \boldsymbol{R}^{F,S}\boldsymbol{\Phi}^S + \boldsymbol{T}^{F,S}\boldsymbol{\Psi}^{F,S} - \boldsymbol{f}^{F,S}. \tag{5.111}$$

- The residuals defined in Equation (5.108) for face $f_{i\mathrm{C}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{C,A}}$ are rewritten in vector form with matrix $\boldsymbol{Q}^{\mathrm{C,A}} \in \mathbb{R}^{n^{\mathrm{C,A}} \times\, n^{\mathrm{B}}}$ composed of rows corresponding to vectors $\boldsymbol{q}_{i\mathrm{C}}$, and identity matrix $\boldsymbol{I}^{\mathrm{C,A}} \in \mathbb{R}^{n^{\mathrm{C,A}} \times n^{\mathrm{C,A}}}$, given as

$$\mathcal{G}^{\mathrm{C,A}}\left(\boldsymbol{\Psi}^{\mathrm{C,A}}, \boldsymbol{\Phi}^{\mathrm{B}}\right) = \boldsymbol{Q}^{\mathrm{C,A}}\boldsymbol{\Phi}^{\mathrm{B}} - \boldsymbol{I}^{\mathrm{C,A}}\boldsymbol{\Psi}^{\mathrm{C,A}}. \tag{5.112}$$

- The residuals defined in Equation (5.109) for face $f_{j\mathrm{C}}$ on computational boundary subset $\Gamma_\Delta^{\mathrm{C,B}}$ are rewritten in vector form with matrix $\boldsymbol{Q}^{\mathrm{C,B}} \in \mathbb{R}^{n^{\mathrm{C,B}} \times n^{\mathrm{A}}}$ composed of rows corresponding to vectors $\boldsymbol{q}_{j\mathrm{C}}$, matrix $\boldsymbol{W}^{\mathrm{C,B}} \in \mathbb{R}^{n^{\mathrm{C,B}} \times n^{\mathrm{C,A}}}$ composed of rows corresponding to vectors $\boldsymbol{w}_{i\mathrm{C}}$, and identity matrix $\boldsymbol{I}^{\mathrm{C,B}} \in \mathbb{R}^{n^{\mathrm{C,B}} \times n^{\mathrm{C,B}}}$, given as

$$\mathcal{G}^{\mathrm{C,B}}\left(\boldsymbol{\Psi}^{\mathrm{C,B}}, \boldsymbol{\Phi}^{\mathrm{B}}, \boldsymbol{\Psi}^{\mathrm{C,A}}\right) = \boldsymbol{Q}^{\mathrm{C,B}}\boldsymbol{\Phi}^{\mathrm{A}} + \boldsymbol{W}^{\mathrm{C,B}}\boldsymbol{\Psi}^{\mathrm{C,A}} - \boldsymbol{I}^{\mathrm{C,B}}\boldsymbol{\Psi}^{\mathrm{C,B}}. \tag{5.113}$$

The explicit formulation consists in gathering in matrix form the residual operators given in Equations (5.110) to (5.113). Then, the system of linear equations from $\mathbb{R}^{DOF}$ to $\mathbb{R}^{DOF}$ is written as

$$\begin{bmatrix} \boldsymbol{R}^{\mathrm{A}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{R}^{\mathrm{R,A}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{R}^{\mathrm{C,A}} & \boldsymbol{0} & \boldsymbol{T}^{\mathrm{C,A}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}^{\mathrm{B}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}^{\mathrm{R,B}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}^{\mathrm{C,B}} & \boldsymbol{0} & \boldsymbol{T}^{\mathrm{C,B}} \\ \boldsymbol{0} & \boldsymbol{Q}^{\mathrm{C,A}} & -\boldsymbol{I}^{\mathrm{C,A}} & \boldsymbol{0} \\ \boldsymbol{Q}^{\mathrm{C,B}} & \boldsymbol{0} & \boldsymbol{W}^{\mathrm{C,B}} & -\boldsymbol{I}^{\mathrm{C,B}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}^{\mathrm{A}} \\ \boldsymbol{\Phi}^{\mathrm{B}} \\ \boldsymbol{\Psi}^{\mathrm{C,A}} \\ \boldsymbol{\Psi}^{\mathrm{C,B}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{A}} \\ \boldsymbol{f}^{\mathrm{R,A}} \\ \boldsymbol{f}^{\mathrm{C,A}} \\ \boldsymbol{f}^{\mathrm{B}} \\ \boldsymbol{f}^{\mathrm{R,B}} \\ \boldsymbol{f}^{\mathrm{C,B}} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{T}^{\mathrm{R,A}}\boldsymbol{\Psi}^{\mathrm{R,A}} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{T}^{\mathrm{R,B}}\boldsymbol{\Psi}^{\mathrm{R,B}} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \tag{5.114}$$

where $\boldsymbol{0}$ indicate null vectors or matrices with the required sizes (vectors $\boldsymbol{\Psi}^{\mathrm{R},S}$ are known and, therefore, are written on the right-hand side). As for the implicit formulation, an iterative free-matrix

algorithm, such as the GMRES method, supplemented with a preconditioning matrix, is used to solve the explicit system of linear equations (5.114).

A static condensation of the explicit formulation is easily derived and provides a system of linear equations in terms of only internal variables. Vectors $\boldsymbol{\Psi}^{\mathrm{C},S}$ are replaced with the corresponding expressions in terms of vectors $\boldsymbol{\Phi}^{S}$, and the condensed explicit formulation is given as

$$\begin{bmatrix} \boldsymbol{R}^{\mathrm{A}} & \boldsymbol{0} \\ \boldsymbol{R}^{\mathrm{R,A}} & \boldsymbol{0} \\ \boldsymbol{R}^{\mathrm{C,A}} & \boldsymbol{T}^{\mathrm{C,A}}\boldsymbol{Q}^{\mathrm{C,A}} \\ \boldsymbol{0} & \boldsymbol{R}^{\mathrm{B}} \\ \boldsymbol{0} & \boldsymbol{R}^{\mathrm{R,B}} \\ \boldsymbol{T}^{\mathrm{C,B}}\boldsymbol{Q}^{\mathrm{C,B}} & \left(\boldsymbol{R}^{\mathrm{C,B}} + \boldsymbol{T}^{\mathrm{C,B}}\boldsymbol{W}^{\mathrm{C,B}}\boldsymbol{Q}^{\mathrm{C,A}}\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}^{\mathrm{A}} \\ \boldsymbol{\Phi}^{\mathrm{B}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{A}} \\ \boldsymbol{f}^{\mathrm{R,A}} \\ \boldsymbol{f}^{\mathrm{C,A}} \\ \boldsymbol{f}^{\mathrm{B}} \\ \boldsymbol{f}^{\mathrm{R,B}} \\ \boldsymbol{f}^{\mathrm{C,B}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{T}^{\mathrm{R,A}}\boldsymbol{\Psi}^{\mathrm{R,A}} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{T}^{\mathrm{R,B}}\boldsymbol{\Psi}^{\mathrm{R,B}} \\ \boldsymbol{0} \end{bmatrix}. \tag{5.115}$$

5.4.4.4 Implementation

The explicit formulation corresponds to linear equations (5.114) or its condensed form (5.115) and is solved with an iterative method. The associated matrix gathers blocks composed of vectors $\boldsymbol{r}_i$, $\boldsymbol{t}_i$, $\boldsymbol{q}_{i\mathrm{C}}$, $\boldsymbol{q}_{j\mathrm{C}}$, and $\boldsymbol{w}_{j\mathrm{C}}$ derived from the global vector mappings applied to vectors of very small size, when compared with the number of variables. Therefore, the associated matrix is very sparse (the same is valid for the condensed form), which is useful from the implementation viewpoint to avoid a substantial number of unnecessary multiplications and save a large amount of memory. Moreover, an explicit formulation with a sparse matrix and a right-hand side vector, replacing the global residual operator, brings advantages over the implicit formulation. Indeed, the number of multiplications and additions becomes smaller, the computational overhead due to loop management in steps 1 to 3 is avoided, and the cache is used more efficiently. Therefore, the overall performance of the proposed method is improved, reducing the execution time of the simulation. Another important advantage is that having the matrix in hand, it turns out more practical to design better preconditioning matrices.

5.5 Numerical benchmark

The verification of the proposed method is based on the numerical analysis in terms of accuracy, convergence order, stability, and robustness to solve conjugate heat transfer problems in arbitrary three-dimensional curved domains. For that purpose, analytic solutions $\phi^{S}(\boldsymbol{x})$ are obtained in associated physical subdomains Ω^{S} such that the continuity interface conditions are fulfilled on the interface subsets. Moreover, functions $g^{\mathrm{R},S}(\boldsymbol{x})$ are obtained from the analytic solutions to prescribe the Robin boundary conditions on physical boundary subsets $\Gamma^{\mathrm{R},S}$. Following the method of manufactured solutions [18–21], source term functions $f^{S}(\boldsymbol{x})$ are determined to satisfy Equation (5.1)

in associated physical subdomains Ω^S given the velocity vector functions $\boldsymbol{u}^S(\boldsymbol{x})$ and the thermal conductivity functions $\kappa^S(\boldsymbol{x})$. Notice that only smooth solutions and source terms of the steady-state three-dimensional conjugate heat transfer problem are considered in the present work. The method of verification consists in running simulations with successive finer polyhedral meshes generated for the curved domains. The approximate solutions computed from the proposed method are compared with the analytic ones and, therefore, the accuracy and convergence orders under mesh refinement are determined.

The system of linear equations is solved in the condensed form with the GMRES method where the number of degrees of freedom is $DOF = n^{\mathrm{A}} + n^{\mathrm{B}}$ corresponding to the approximate cell mean-values of the temperature functions. For each cell c_i in computational subdomain Ω^S_Δ, the error of approximate cell mean-value ϕ^*_i is determined having the exact cell mean-value given as

$$\overline{\phi}_i = \frac{1}{|c_i|}\int_{c_i} \phi^S(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{5.116}$$

Then, the relative errors in the L^1- and L^∞-norm, denoted as E_1 and E_∞, respectively, are determined as

$$E_1 = \frac{\sum_{i=1}^{DOF} \left|\phi^*_i - \overline{\phi}_i\right||c_i|}{\sum_{i=1}^{DOF} |c_i|}, \tag{5.117}$$

$$E_\infty = \max_{i=1}^{DOF} \left|\phi^*_i - \overline{\phi}_i\right|. \tag{5.118}$$

Consider two meshes for the same domain with different characteristic sizes, a number of degrees of freedom of DOF_1 and DOF_2, respectively, associated relative errors in the L^1-norm of $E_{1,1}$ and $E_{1,2}$, respectively, and associated relative errors in the L^∞-norm of $E_{\infty,1}$ and $E_{\infty,2}$, respectively. Then, the convergence orders for the relative errors in the L^1- and L^∞-norms, denoted as O_1 and O_∞, respectively, are given as

$$O_1 = 3\left|\frac{\ln\left(E_{1,1}/E_{1,2}\right)}{\ln(DOF_1/DOF_2)}\right|, \tag{5.119}$$

$$O_\infty = 3\left|\frac{\ln\left(E_{\infty,1}/E_{\infty,2}\right)}{\ln(DOF_1/DOF_2)}\right|. \tag{5.120}$$

In the case of the continuity interface conditions, the partitioning method prescribes either a Dirichlet boundary condition in one subproblem and a Neumann boundary condition in the other, resulting in two possible partitioning configurations (see Remark 21). These two partitioning configurations are referred to as configurations DN and ND and are defined as follows:

- DN: a Neumann boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,A}}$ in subproblem P^{A} while a Dirichlet boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,B}}$ in subproblem P^{B}.
- ND: a Dirichlet boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,A}}$ in subproblem P^{A} while a Neumann boundary condition is prescribed on physical interface subset $\Gamma^{\mathrm{C,B}}$ in subproblem P^{B}.

Specific test cases are addressed with both partitioning configurations, and the obtained approximate solutions are compared in terms of accuracy and convergence orders, establishing their behaviour with respect to the physical properties of the problem.

The weighting function to perform the minimization procedure in the polynomial reconstruction method is set with parameters $\delta = 2$ and $\sigma = 5h$, where h is the characteristic size of the reference mesh element (face or cell). The optimal parameters are chosen to provide stability and robustness to the least-squares method.

5.5.1 Spherical shell domain test case

The first test case consists of a spherical shell with two layers and spherical boundaries and interface centred at the axis origin (see Figure 5.2). The inner layer corresponds to physical subdomain Ω^{A} and the outer layer corresponds to physical subdomain Ω^{B}. The internal and external spherical surfaces have radius $r_{\mathrm{I}} = 0.5$ and $r_{\mathrm{E}} = 1$, respectively, while the middle spherical surface has radius $r_{\mathrm{M}} = 0.75$.

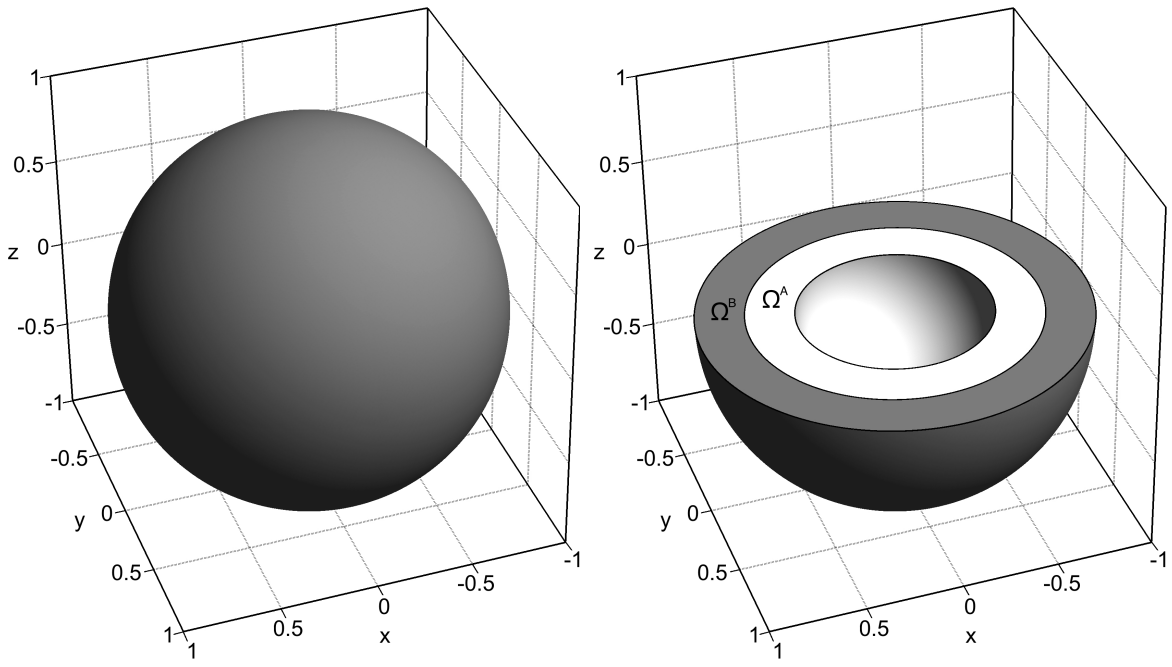
The analytic solutions assigned to this test case are expressed in spherical coordinates (r, θ, φ), with radius $r^2 = x^2 + y^2 + z^2$, polar angle $\theta = \arccos(z/r)$, and azimuthal angle $\theta = \arctan(y/x)$, given as

$$\phi^S(r,\theta,\varphi) = \left(\frac{a^S}{r} + b^S\right)\sin\left(n_\theta^S\theta\right)\sin^5(\theta)\cos\left(n_\varphi^S\varphi\right), \quad \text{in } \Omega^S, \tag{5.121}$$

for $S \in \{\mathrm{A},\mathrm{B}\}$, with parameters $a^{\mathrm{A}}, b^{\mathrm{A}}, a^{\mathrm{B}}, b^{\mathrm{B}} \in \mathbb{R}$ and $n_\theta^{\mathrm{A}}, n_\varphi^{\mathrm{A}}, n_\theta^{\mathrm{B}}, n_\varphi^{\mathrm{B}} \in \mathbb{N}$. Notice that, although the analytic solutions are provided in spherical coordinates, the problem is numerically solved in Cartesian coordinates. The associated source term functions are obtained after substituting analytic solutions (5.121) into Equation (5.1).

The internal and external surfaces of physical subdomains Ω^{A} and Ω^{B}, respectively, correspond to physical boundary subsets $\Gamma^{\mathrm{R,A}}$ and $\Gamma^{\mathrm{R,B}}$, respectively. Physical boundary subset $\Gamma^{\mathrm{R,A}}$ is assigned with an homogeneous Dirichlet boundary condition, while physical boundary subset $\Gamma^{\mathrm{R,B}}$ is assigned with a periodic Dirichlet boundary condition with boundary condition function given as

$$g^{\mathrm{R,B}}(\theta,\varphi) = \left(\frac{a^{\mathrm{B}}}{r_{\mathrm{E}}} + b^{\mathrm{B}}\right)\sin\left(n_\theta^{\mathrm{B}}\theta\right)\sin^5(\theta)\cos\left(n_\varphi^{\mathrm{B}}\varphi\right), \quad \text{on } \Gamma^{\mathrm{D,B}}, \tag{5.122}$$



(a) Full view. (b) Cross-section view.

Figure 5.2: Physical domain for the spherical shell domain test case.

where coefficient functions $\alpha^{\mathrm{R,A}}(\boldsymbol{x}) = 1$, $\beta^{\mathrm{R,A}}(\boldsymbol{x}) = 0$, $\alpha^{\mathrm{R,B}}(\boldsymbol{x}) = 1$, and $\beta^{\mathrm{R,B}}(\boldsymbol{x}) = 0$ are assigned on the latter such that the Robin boundary conditions corresponds to the Dirichlet one.

The middle surface between physical subdomains Ω^{A} and Ω^{B} corresponds to physical interface subset Γ^{C}, where the continuity interface conditions are prescribed. Parameters $a^{\mathrm{A}}, a^{\mathrm{B}}, b^{\mathrm{A}}, b^{\mathrm{B}}$ are determined such that the boundary conditions and the continuity interface conditions are simultaneously satisfied. Constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x})$ and $\kappa^{\mathrm{B}}(\boldsymbol{x})$ are assigned and parameters are determined as

$$a^{\mathrm{A}} = c\kappa^{\mathrm{B}} r_{\mathrm{I}} r_{\mathrm{M}} r_{\mathrm{E}}, \tag{5.123}$$

$$a^{\mathrm{B}} = c\kappa^{\mathrm{A}} r_{\mathrm{I}} r_{\mathrm{M}} r_{\mathrm{E}}, \tag{5.124}$$

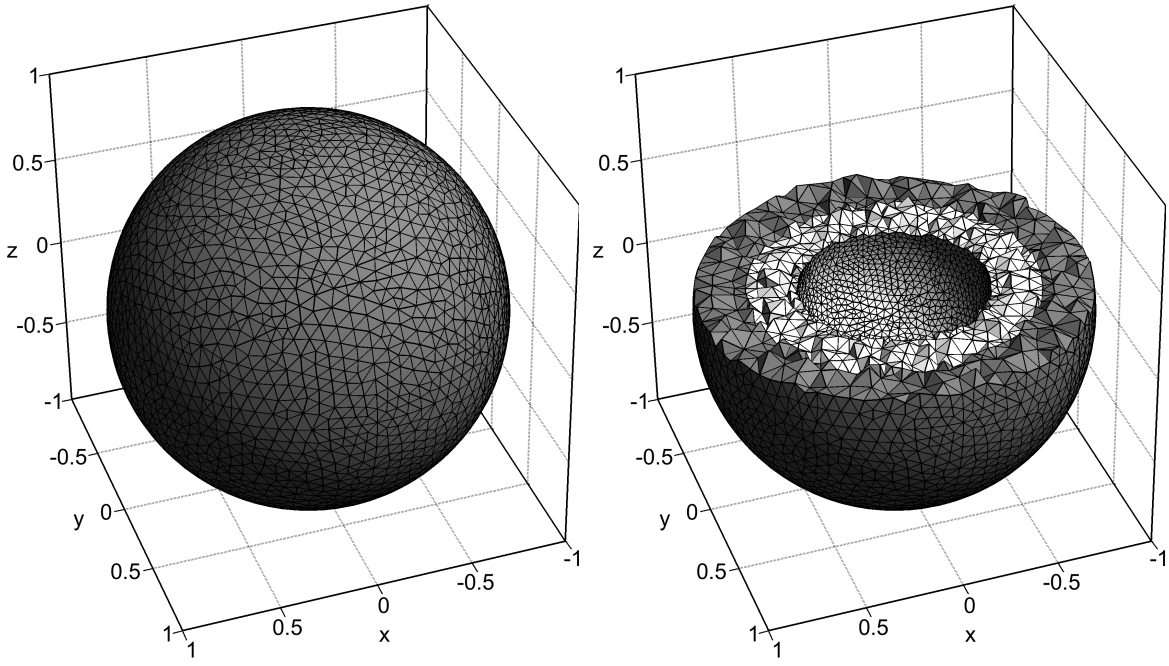
$$b^{\mathrm{A}} = -c\kappa^{\mathrm{B}} r_{\mathrm{M}} r_{\mathrm{E}}, \tag{5.125}$$

$$b^{\mathrm{B}} = -cr_{\mathrm{E}}\Big(\kappa^{\mathrm{A}} r_{\mathrm{I}} + \kappa^{\mathrm{B}}(r_{\mathrm{M}} - r_{\mathrm{I}})\Big), \tag{5.126}$$

$$c = \kappa^{\mathrm{A}}(r_{\mathrm{I}} r_{\mathrm{M}} - r_{\mathrm{I}} r_{\mathrm{E}}) + \kappa^{\mathrm{B}}(r_{\mathrm{I}} r_{\mathrm{E}} - r_{\mathrm{M}} r_{\mathrm{E}}). \tag{5.127}$$

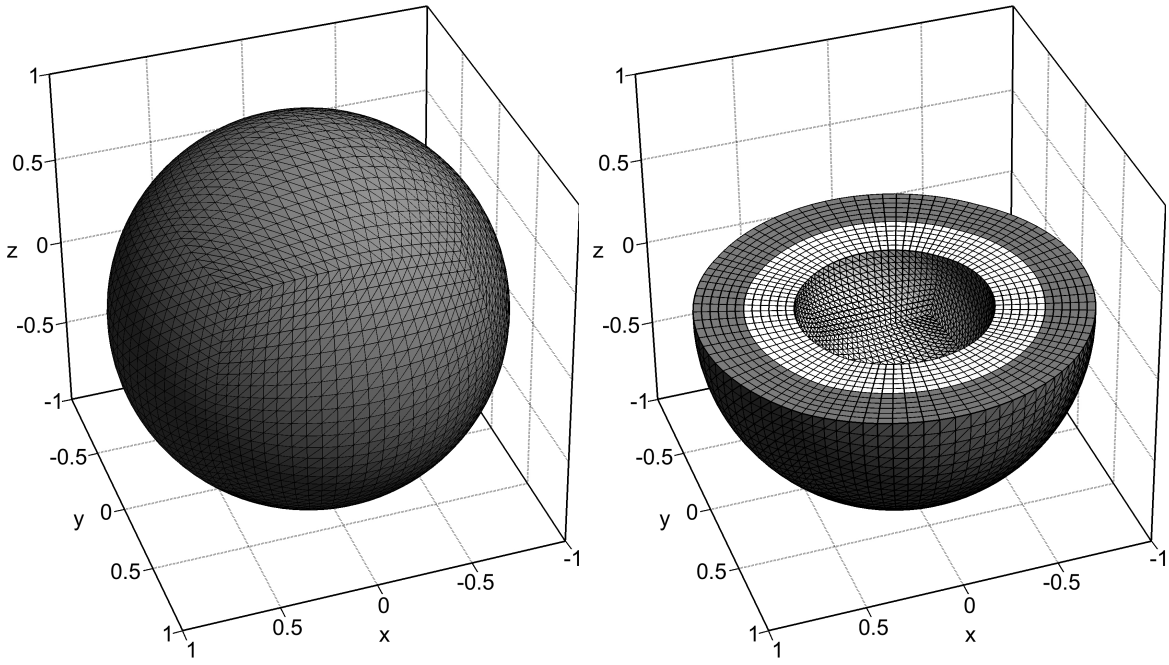
Density and heat capacity functions are assigned constant while the velocity vector fields are tangent to the surfaces such that no convection occurs through the boundaries and interface, given in spherical coordinates as

$$\boldsymbol{u}^S(r,\theta) = \omega^S r \sin(\theta)\hat{\boldsymbol{\varphi}}, \quad \text{in } \Omega^S, \tag{5.128}$$



(a) Full view. (b) Cross-section view.

Figure 5.3: Uniform Delaunay tetrahedral meshes for the spherical shell domain test case.



(a) Full view. (b) Cross-section view.

Figure 5.4: Uniform structured hexahedral meshes for the spherical shell domain test case.

for $S \in \{\mathrm{A}, \mathrm{B}\}$, with parameters $\omega^{\mathrm{A}}, \omega^{\mathrm{B}} \in \mathbb{R}$.

The test case is addressed with parameters $n_\theta^{\mathrm{A}} = n_\varphi^{\mathrm{A}} = n_\theta^{\mathrm{B}} = n_\varphi^{\mathrm{B}} = 5$, $\rho^{\mathrm{A}} = \rho^{\mathrm{B}} = 1$, $c_{\mathrm{p}}^{\mathrm{A}} = c_{\mathrm{p}}^{\mathrm{B}} = 1$. The analytic solutions only depend on these parameters and on the thermal conductivity functions (through parameters a^{A}, b^{A}, a^{B}, and b^{B}), which are assigned according to the following cases:

- Low thermal conductivity ratio case: $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 1$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 2$ (see Figure 5.5).
- High thermal conductivity ratio case: $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 1$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 100$ (see Figure 5.6),

where the ratio is defined as $\kappa^{\mathrm{B}}(\boldsymbol{x})/\kappa^{\mathrm{A}}(\boldsymbol{x})$. Parameters ω^{A} and ω^{B} in the velocity functions are assigned according to the following cases:

- Low thermal convection case: $\omega^{\mathrm{A}} = 1$ and $\omega^{\mathrm{B}} = -1$.
- High thermal convection case: $\omega^{\mathrm{A}} = 100$ and $\omega^{\mathrm{B}} = -100$.

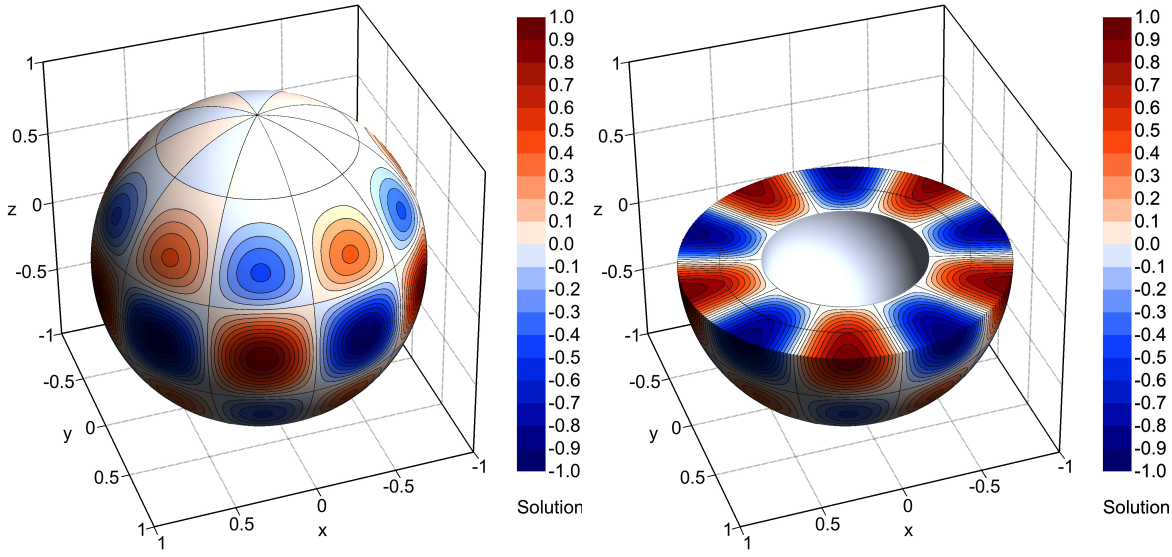
Successively finer uniform Delaunay tetrahedral meshes or uniform structured prismatic meshes (with no curved elements) are used to discretize each physical subdomain (see Figure 5.3 and 5.4) and the simulations are carried out for configurations DN and ND and for polynomial degrees $d = 1, 3, 5$.

5.5.1.1 Low thermal convection case

The low thermal convection case combined with the low and high thermal conductivity ratio cases are addressed, and simulations are carried out with successive finer uniform Delaunay tetrahedral meshes, polynomial degrees $d = 1, 3, 5$, and configurations DN and ND. The relative errors and convergence orders are reported in Table 5.4. Notice that configuration DN prescribes the Neumann boundary condition in subproblem P^{A} (with the highest thermal conductivity), while the Dirichlet boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity). In contrast, configuration ND prescribes the opposite situation.

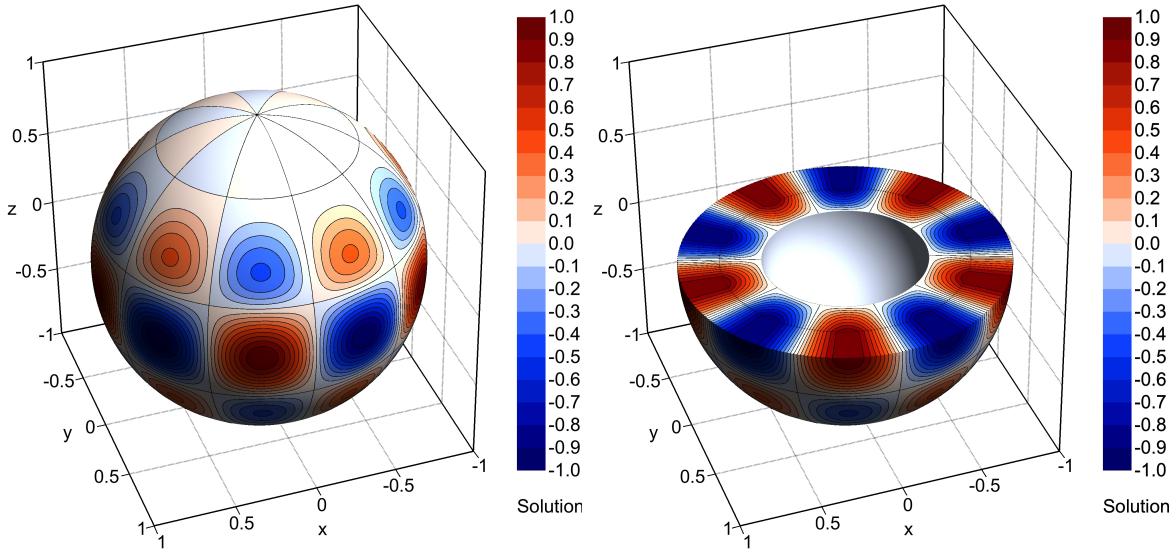
In both thermal conductivity ratio cases, and regardless of the partitioning configuration, the second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms. Additionally, the accuracy obtained with configurations DN and ND is comparable for the same case and polynomial degree, whereas the difference seems to reduce for lower thermal conductivity ratios. On the other side, the thermal conductivity ratio has a low influence on the accuracy, which supports that the proposed method is robust and stable regardless of the strength of the thermal interaction between physical subdomains.

Although the partitioning configuration seems of minor importance in what concerns the accuracy and the convergence orders, the same comparative analysis is carried out in terms of the residual convergence in the GMRES method to solve the associated system of linear equations. The GMRES stopping criteria corresponds to a tolerance parameter for the residual convergence assigned



(a) Full view. (b) Cross-section view.

Figure 5.5: Analytic solution for the spherical shell domain test case with a low thermal conductivity ratio.



(a) Full view. (b) Cross-section view.

Figure 5.6: Analytic solution for the spherical shell domain test case with a high thermal conductivity ratio.

according to the characteristic mesh size and the polynomial degree, such that unnecessarily lower residuals, for the expected approximate solution accuracy, are avoided. Additionally, a maximum number of 200 000 iterations is also set.

The number of GMRES iterations is reported in Table 5.5. In the low thermal conductivity ratio case, the number of iterations is comparable between configurations DN and ND regardless of the polynomial degree. However, the former requires, in general, slightly fewer iterations. Similarly, in the high thermal conductivity ratio case, configuration DN requires fewer iterations than configuration ND, regardless of the polynomial degree. However, in that case, the difference is more considerable for polynomial degree $d = 5$ and increases as the mesh characteristic size decreases. Configuration ND requires more iterations, possibly due to the higher condition number of the associated coefficients matrices. Since both partitioning configurations provide comparable results in terms of accuracy and convergence order, configuration DN performs more efficiently than configuration ND. Therefore, based on the previous results, the Dirichlet and Neumann boundary conditions resulted from the partitioning method for the continuity interface condition, should be prescribed in the subproblem with the lower and higher thermal conductivity, respectively.

The relative errors as a function of the associated solution time, denoted as T_{S} (in seconds (s)), are plotted in Figure 5.7. As observed, the solution time for the same mesh increases for increasing polynomial degree since the sizes of the matrices and vectors involved in the polynomial reconstructions are higher. On the other side, to achieve the same accuracy, the fourth-order of convergence is always significantly faster than the second-order of convergence and, in turn, the sixth-order of convergence is substantially faster than the former. For instance, to achieve a relative error of $E_1 = 10^{-6}$ in the L^1-norm, the second-, fourth-, and sixth-orders of convergence require approximately 10^7 s, 100 s, and 10 s, respectively. For the second-order of convergence, roughly four months are required (extrapolating the associated relative error curve), while only a few seconds are needed with the fourth- and sixth-orders of convergence. Moreover, to achieve such accuracy, the second-order of convergence also implies to generate and store massive mesh composed of billions to trillions of cells, for which the amount of memory exceeds the nowadays computer limitations. Indeed, the benefit of using very high-order of convergence is to achieve the same accuracy as second-order of convergence but with substantially coarser meshes and significantly faster simulations.

Structured meshes are useful for specific situations, and different mesh elements than tetrahedrons can be used. The same test case is addressed with successive finer uniform structured prismatic meshes to check whether the proposed method can handle other types of polyhedral meshes. The relative errors and convergence orders are reported in Table 5.6.

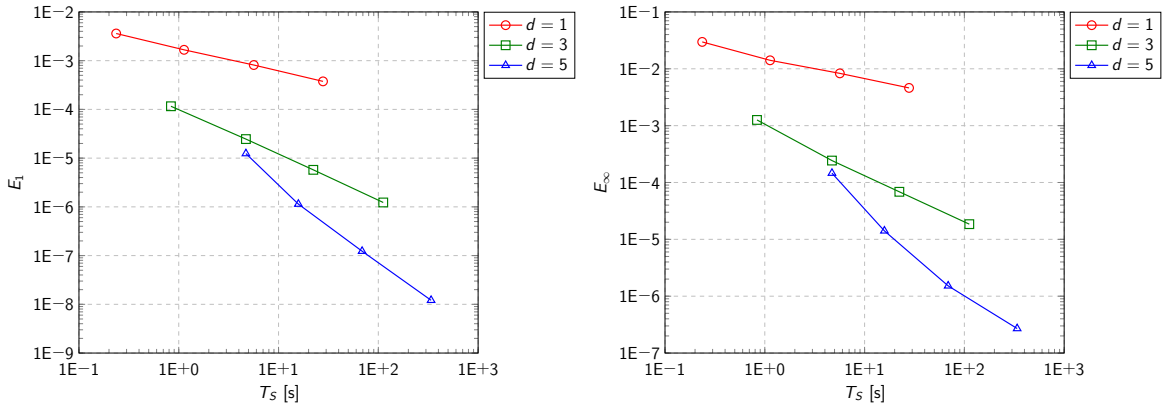
As observed, in both thermal conductivity ratio cases, the second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms, with configuration DN. Therefore, the results support the capability of the proposed method to handle structured meshes composed of triangular prisms. The results obtained with configuration ND are not provided for the sake of simplicity, but the same behaviour as for uniform

Table 5.4: Relative errors and convergence orders obtained in the spherical shell domain test case with a low thermal convection and uniform Delaunay tetrahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio, configuration DN												
80 829	3.61E−03	–	2.97E−02	–	1.16E−04	–	1.26E−03	–	1.24E−05	–	1.46E−04	–
254 516	1.67E−03	2.01	1.41E−02	1.95	2.46E−05	4.06	2.43E−04	4.30	1.14E−06	6.23	1.41E−05	6.13
726 929	8.14E−04	2.06	8.31E−03	1.51	5.75E−06	4.16	6.86E−05	3.62	1.23E−07	6.38	1.52E−06	6.36
2 290 543	3.76E−04	2.02	4.61E−03	1.54	1.23E−06	4.03	1.85E−05	3.43	1.21E−08	6.05	2.70E−07	4.52
Low thermal conductivity ratio, configuration ND												
80 829	3.41E−03	–	2.84E−02	–	1.11E−04	–	1.22E−03	–	1.17E−05	–	1.51E−04	–
254 516	1.58E−03	2.01	1.45E−02	1.75	2.33E−05	4.09	2.79E−04	3.86	1.07E−06	6.25	1.44E−05	6.15
726 929	7.68E−04	2.07	8.32E−03	1.59	5.40E−06	4.18	7.19E−05	3.88	1.15E−07	6.37	1.52E−06	6.44
2 290 543	3.55E−04	2.02	4.34E−03	1.70	1.15E−06	4.04	1.87E−05	3.53	1.13E−08	6.07	2.74E−07	4.47
High thermal conductivity ratio, configuration DN												
80 829	4.42E−03	–	4.68E−02	–	1.70E−04	–	2.34E−03	–	2.06E−05	–	2.83E−04	–
254 516	2.04E−03	2.03	1.91E−02	2.34	3.61E−05	4.06	4.91E−04	4.08	1.95E−06	6.17	3.23E−05	5.68
726 929	9.85E−04	2.08	1.09E−02	1.60	8.36E−06	4.18	1.49E−04	3.40	2.09E−07	6.37	3.41E−06	6.43
2 290 543	4.56E−04	2.01	6.03E−03	1.56	1.79E−06	4.03	3.16E−05	4.06	2.15E−08	5.95	4.64E−07	5.21
High thermal conductivity ratio, configuration ND												
80 829	3.56E−03	–	3.21E−02	–	1.32E−04	–	1.48E−03	–	1.58E−05	–	2.16E−04	–
254 516	1.63E−03	2.04	1.63E−02	1.78	2.75E−05	4.10	3.75E−04	3.59	1.51E−06	6.14	2.51E−05	5.63
726 929	7.85E−04	2.09	9.46E−03	1.55	6.26E−06	4.23	8.86E−05	4.12	1.61E−07	6.40	2.16E−06	7.00
2 290 543	3.61E−04	2.03	4.94E−03	1.70	1.32E−06	4.07	2.09E−05	3.77	1.59E−08	6.04	3.44E−07	4.80

Table 5.5: Number of GMRES iterations obtained in the spherical shell domain test case with a low thermal convection and uniform Delaunay tetrahedral meshes.

	Low thermal conductivity ratio						High thermal conductivity ratio					
	Configuration DN			Configuration ND			Configuration DN			Configuration ND		
DOF	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$
80 829	32	72	192	37	79	137	29	51	74	37	80	384
254 516	46	106	206	50	119	213	42	82	106	51	119	453
726 929	63	153	281	68	170	310	60	123	149	73	173	5 468
2 290 543	97	242	423	104	269	402	91	191	212	108	268	67 512



(a) Relative error in the L^1-norm.

(b) Relative error in the L^∞-norm.

Figure 5.7: Relative errors and solution time obtained in the spherical shell domain test case with a low thermal convection, low thermal conductivity ratio, configuration DN, and uniform Delaunay tetrahedral meshes.

Delaunay tetrahedral meshes is observed.

Table 5.6: Relative errors and convergence orders obtained in the spherical shell domain test case with a low thermal convection, configuration DN, and uniform structured prismatic meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio												
76 176	3.44E−03	–	2.37E−02	–	1.52E−04	–	1.28E−03	–	4.57E−05	–	6.79E−04	–
235 224	1.63E−03	1.99	1.14E−02	1.95	3.82E−05	3.67	3.07E−04	3.80	4.72E−06	6.04	1.25E−04	4.51
710 976	7.80E−04	2.00	5.35E−03	2.05	9.59E−06	3.75	7.15E−05	3.95	5.31E−07	5.92	1.57E−05	5.61
2 154 720	3.87E−04	1.89	2.67E−03	1.88	2.22E−06	3.96	1.76E−05	3.79	6.36E−08	5.74	3.05E−06	4.43
High thermal conductivity ratio												
76 176	4.19E−03	–	3.19E−02	–	2.23E−04	–	2.38E−03	–	8.15E−05	–	1.46E−03	–
235 224	2.00E−03	1.97	1.55E−02	1.92	5.45E−05	3.75	5.89E−04	3.72	7.87E−06	6.22	1.55E−04	5.97
710 976	9.42E−04	2.04	6.54E−03	2.34	1.34E−05	3.81	1.43E−04	3.85	8.96E−07	5.89	1.95E−05	5.62
2 154 720	4.78E−04	1.83	3.28E−03	1.87	3.08E−06	3.98	3.14E−05	4.09	1.04E−07	5.82	3.79E−06	4.43

5.5.1.2 High thermal convection case

The high thermal convection case combined with the low and high thermal conductivity ratio cases are addressed, and simulations are carried out with successive finer uniform Delaunay tetrahedral meshes, polynomial degrees $d = 1, 3, 5$, and configurations DN and ND. The relative errors and convergence orders are reported in Table 5.7. Notice that configuration DN prescribes the Neumann boundary condition in subproblem P^{A} (with the highest thermal conductivity), while the Dirichlet boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity). In contrast, configuration ND prescribes the opposite situation.

As in the previous case, in both thermal conductivity ratios cases and regardless of the partitioning configuration, the second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms. The accuracy is also comparable between configurations DN and ND and has a low dependency on the strength of the thermal interaction between physical subdomains. Moreover, the results preserve the essentially non-oscillatory behaviour with any polynomial degree and the when thermal convection increases, when compared to the previous case. Comparing the relative errors in the L^1- and L^∞-norms for the same approximate solution, a ratio of around ten is obtained, regardless of the polynomial degree. Such ratio indicates that the relative error distribution is relatively even in the physical subdomains and that the interface treatment is not deteriorating the accuracy, either locally or globally. The results support that the proposed method is robust and stable with very high-order of convergence for convection-dominant cases.

The number of GMRES iterations are reported in Table 5.8. More iterations are required in comparison with the low thermal convection case, possibly due to the increased condition number of the associated matrices of coefficients. In the low thermal conductivity ratio case, the number of iterations is comparable between configurations DN and ND regardless of the polynomial degree. However, in general, a few more iterations are required for the latter configuration, as observed previously. On the other side, polynomial degree $d = 5$ also requires a considerably higher number of iterations with configuration ND than with configuration DN, indicating that higher condition numbers are involved, and the latter partitioning configuration is less stable. The results support the previous conclusions concerning the performance of the proposed method with the different partitioning configurations. That is, although comparable accuracy and convergence orders are obtained, regardless of the partitioning configuration, the resulting Dirichlet boundary condition should be prescribed in the subproblem with the lower thermal conductivity (and the resulting Neumann boundary condition in the other subproblem). Such configuration avoids a significant performance deterioration in terms of residual convergence in the GMRES method.

Remark 24 One simulation attains the maximum number of iterations without achieving the tolerance parameter for the residual convergence. In that case, the approximate solution is computed since a substantially higher number of iterations (>200 000) would be required, therefore becoming very time-consuming. However, with the purpose to determine the corresponding relative errors reported in Table 5.4, the approximate solution is computed providing an initial guess to the GMRES method that corresponds to the exact solution. Since the exact and approximate solutions are relatively close, the residual convergence is substantially faster. Still, the obtained number of iterations can not be taken into account for the analysis since such initial guess is not available in practical problems.

The same test case is addressed with successive finer uniform structured prismatic meshes and the relative errors and convergence orders are reported in Table 5.9.

In both thermal conductivity ratio cases, the second-, fourth-, and sixth-orders of convergence

Table 5.7: Relative errors and convergence orders obtained in the spherical shell domain test case with a high thermal convection and uniform Delaunay tetrahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio, configuration DN												
80 829	2.27E−03	–	2.42E−02	–	8.88E−05	–	9.75E−04	–	1.12E−05	–	1.29E−04	–
254 516	1.06E−03	2.00	1.10E−02	2.06	1.81E−05	4.16	2.25E−04	3.84	9.88E−07	6.36	1.51E−05	5.63
726 929	5.30E−04	1.98	6.90E−03	1.34	4.29E−06	4.12	5.72E−05	3.91	1.07E−07	6.35	1.38E−06	6.83
2 290 543	2.53E−04	1.94	3.38E−03	1.87	9.46E−07	3.95	1.43E−05	3.62	1.12E−08	5.90	2.60E−07	4.36
Low thermal conductivity ratio, configuration DN												
80 829	2.39E−03	–	2.26E−02	–	9.01E−05	–	8.77E−04	–	1.11E−05	–	1.33E−04	–
254 516	1.12E−03	2.00	1.09E−02	1.91	1.78E−05	4.25	2.69E−04	3.09	9.27E−07	6.50	1.18E−05	6.33
726 929	5.58E−04	1.98	6.81E−03	1.34	4.14E−06	4.16	5.61E−05	4.49	1.00E−07	6.36	1.42E−06	6.07
2 290 543	2.63E−04	1.96	3.35E−03	1.85	9.01E−07	3.99	1.42E−05	3.59	9.97E−09	6.03	2.57E−07	4.46
High thermal conductivity ratio, configuration DN												
80 829	3.83E−03	–	4.41E−02	–	1.47E−04	–	2.20E−03	–	1.84E−05	–	2.55E−04	–
254 516	1.78E−03	2.00	1.81E−02	2.34	3.14E−05	4.04	4.43E−04	4.19	1.76E−06	6.14	3.14E−05	5.48
726 929	8.65E−04	2.06	1.02E−02	1.65	7.28E−06	4.17	1.41E−04	3.28	1.89E−07	6.37	3.25E−06	6.48
2 290 543	4.04E−04	1.99	5.03E−03	1.84	1.58E−06	3.99	3.04E−05	4.00	2.11E−08	5.74	4.49E−07	5.18
High thermal conductivity ratio, configuration ND												
80 829	3.38E−03	–	3.20E−02	–	1.28E−04	–	1.48E−03	–	1.56E−05	–	2.22E−04	–
254 516	1.55E−03	2.03	1.62E−02	1.78	2.65E−05	4.12	3.75E−04	3.58	1.46E−06	6.18	2.51E−05	5.71
726 929	7.49E−04	2.09	9.39E−03	1.57	6.02E−06	4.24	8.84E−05	4.13	1.56E−07	6.41	2.16E−06	7.01
2 290 543	3.46E−04	2.02	4.93E−03	1.69	1.28E−06	4.04	2.08E−05	3.79	1.56E−08	6.02	3.44E−07	4.81

Table 5.8: Number of GMRES iterations obtained in the spherical shell domain test case with a high thermal convection and uniform Delaunay tetrahedral meshes.

	Low thermal conductivity ratio						High thermal conductivity ratio					
	Configuration DN			Configuration ND			Configuration DN			Configuration ND		
DOF	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$
80 829	69	142	312	70	155	232	61	119	167	69	148	887
254 516	103	244	367	105	261	370	100	205	261	111	256	4 468
726 929	154	387	619	156	410	582	145	324	389	161	409	69 429
2 290 543	237	670	955	246	722	921	236	529	579	263	685	>200 000

are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms, with configuration DN. On the other side, configuration ND provides the behaviour observed previously. As supported by the previous results, the proposed method is capable of handling a three-dimensional domain successfully with curved boundaries and interface. Moreover, the method shows no specific limitations in treating unstructured and structured meshes, conduction and convection dominant cases, low and high thermal conductivity ratio cases, and always provides a very high-order of convergence.

Table 5.9: Relative errors and convergence orders obtained in the spherical shell domain test case with a high thermal convection, configuration DN, and uniform structured prismatic meshes.

	$d = 1$				$d = 3$				$d = 5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio												
76 176	1.44E−03	–	1.53E−02	–	1.03E−04	–	7.78E−04	–	3.92E−05	–	6.93E−04	–
235 224	7.42E−04	1.77	8.28E−03	1.63	2.63E−05	3.63	2.44E−04	3.08	3.88E−06	6.16	1.19E−04	4.69
710 976	3.98E−04	1.69	4.33E−03	1.76	7.07E−06	3.56	6.36E−05	3.65	4.19E−07	6.03	1.57E−05	5.50
2 154 720	2.11E−04	1.72	2.17E−03	1.87	1.71E−06	3.85	1.62E−05	3.70	5.20E−08	5.65	2.85E−06	4.62
High thermal conductivity ratio												
76 176	3.47E−03	–	2.76E−02	–	2.02E−04	–	2.34E−03	–	6.87E−05	–	1.21E−03	–
235 224	1.67E−03	1.93	1.35E−02	1.90	5.12E−05	3.66	5.83E−04	3.69	7.31E−06	5.96	1.49E−04	5.58
710 976	8.05E−04	1.99	6.02E−03	2.19	1.28E−05	3.75	1.42E−04	3.83	8.42E−07	5.86	1.95E−05	5.51
2 154 720	4.12E−04	1.81	3.11E−03	1.79	2.97E−06	3.96	3.12E−05	4.10	9.86E−08	5.80	3.54E−06	4.61

5.5.2 Wavy sleeve domain test case

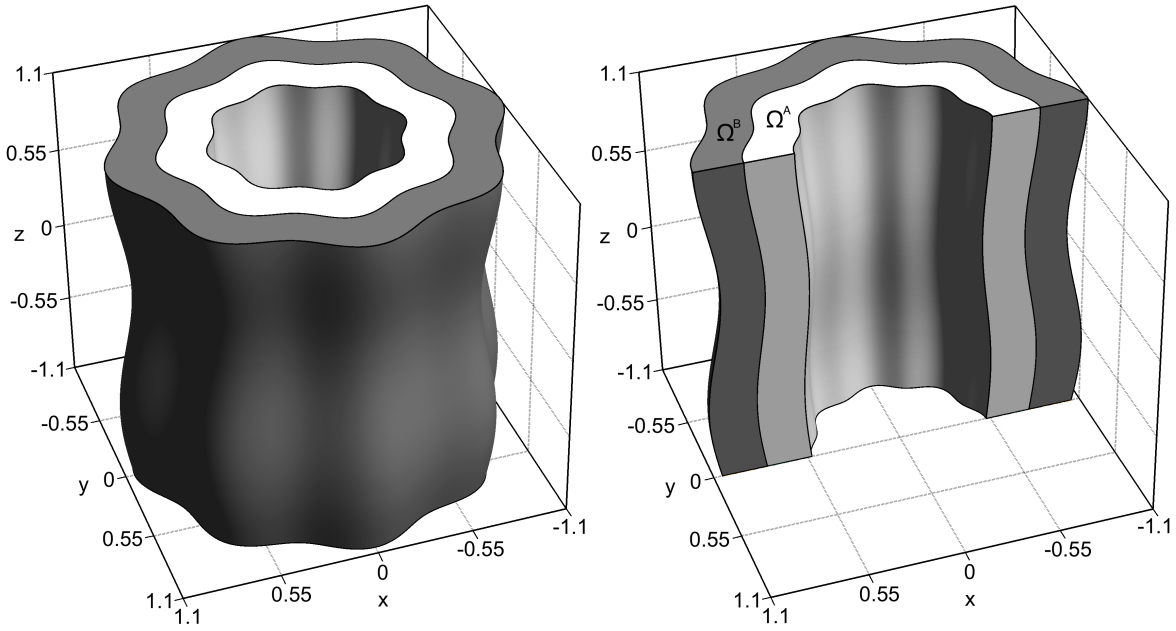
The second test case consists of a non-trivial parameterized geometry that resembles a wavy sleeve with two layers and non-trivial parameterized curved boundaries and interface centred at the axis origin (see Figure 5.8). The domain is obtained applying a diffeomorphic transformation to a cylindrical shell with two layers centered at the axis origin, defined as $(r', \theta', z') \in \Omega' = \{r' \in [r_\mathrm{I}, r_\mathrm{E}], \theta' \in [0, 2\pi], z' \in [z_\mathrm{B}, z_\mathrm{T}]\}$. The inner layer corresponds to physical subdomain Ω^A, the outer layer corresponds to physical subdomain Ω^B, the lateral internal and external surfaces of the cylinder have radius $r_\mathrm{I} = 0.5$ and $r_\mathrm{E} = 1$, the milddle surface has radius $r_\mathrm{M} = 0.75$, and the bottom and top surfaces have applicates $z_\mathrm{B} = -0.25$ and $z_\mathrm{T} = 0.25$, respectively. The diffeomorphic transformation consists in a periodic mapping from domain Ω' onto domain Ω given in cylindrical coordinates as

$$\Omega' \rightarrow \Omega : \begin{bmatrix} r' \\ \theta' \\ z' \end{bmatrix} \rightarrow \begin{bmatrix} r \\ \theta \\ z \end{bmatrix} = \begin{bmatrix} T(r', \theta', z') \\ \theta' \\ z' \end{bmatrix}, \tag{5.129}$$

where function $T(r', \theta', z')$ is a mapping with parameters $\alpha_\theta, \beta_\theta, \alpha_z, \beta_z \in \mathbb{R}$ that converts radius $r' \in \Omega'$ into radius $r \in \Omega$, given as

$$T(r', \theta', z') = r'\Big(1 + \beta_\theta \cos(\alpha_\theta \theta')\Big)\left(1 + \beta_z \sin\left(\alpha_z\left(\frac{z' - z_\mathrm{B}}{z_\mathrm{T} - z_\mathrm{B}}\right)\right)\right). \tag{5.130}$$

For a reversed mapping, function $T^{-1}(r, \theta, z)$ converts a given radius $r \in \Omega$ into a radius $r' \in \Omega'$.



(a) Full view. (b) Cross-section view.

Figure 5.8: Physical domain for the wavy sleeve domain test case.

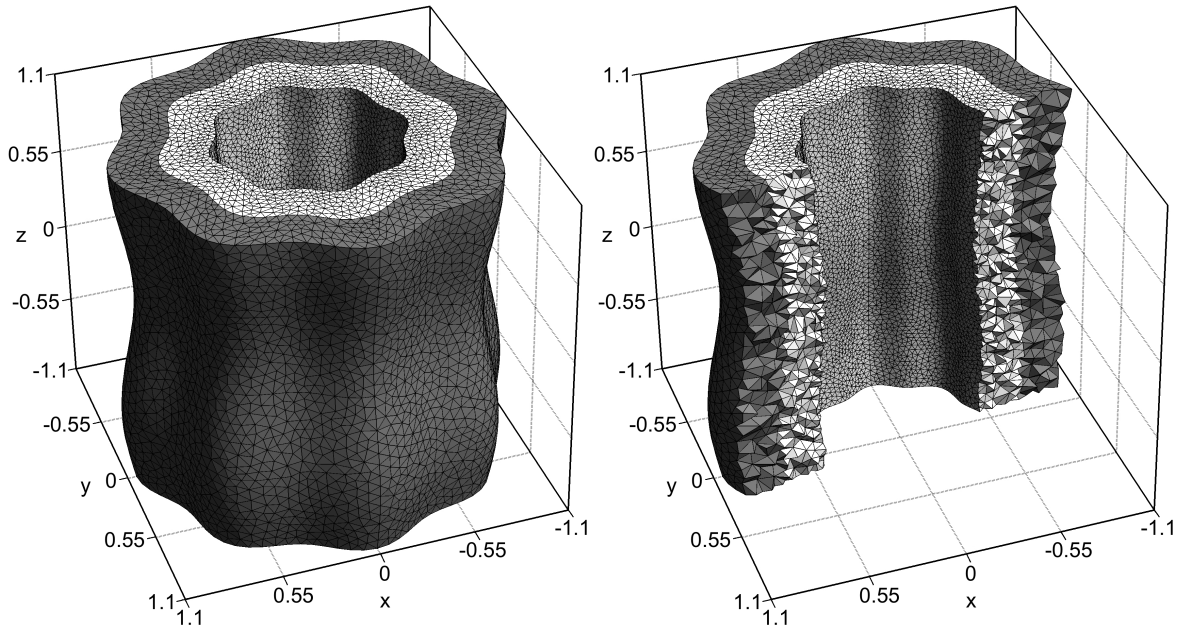
Unit normal vector function on the boundaries and interface, $\boldsymbol{n}^S := \boldsymbol{n}^S(\theta, z)$, are obtained applying the chain rule to function $T(r, \theta, z)$ and is given in cylindrical coordinates as

$$\boldsymbol{n}^S(\theta, z) = \pm\frac{1}{c}\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} T(r_*, \theta, z) \\ \partial_\theta T(r_*, \theta, z) \\ T(r_*, \theta, z)\partial_z T(r_*, \theta, z) \end{bmatrix}, \quad \text{on } \Gamma^S, \tag{5.131}$$

for $S \in \{\mathrm{A}, \mathrm{B}\}$, where the resulting vector is expressed in Cartesian coordinates and c is the normalization constant, given as

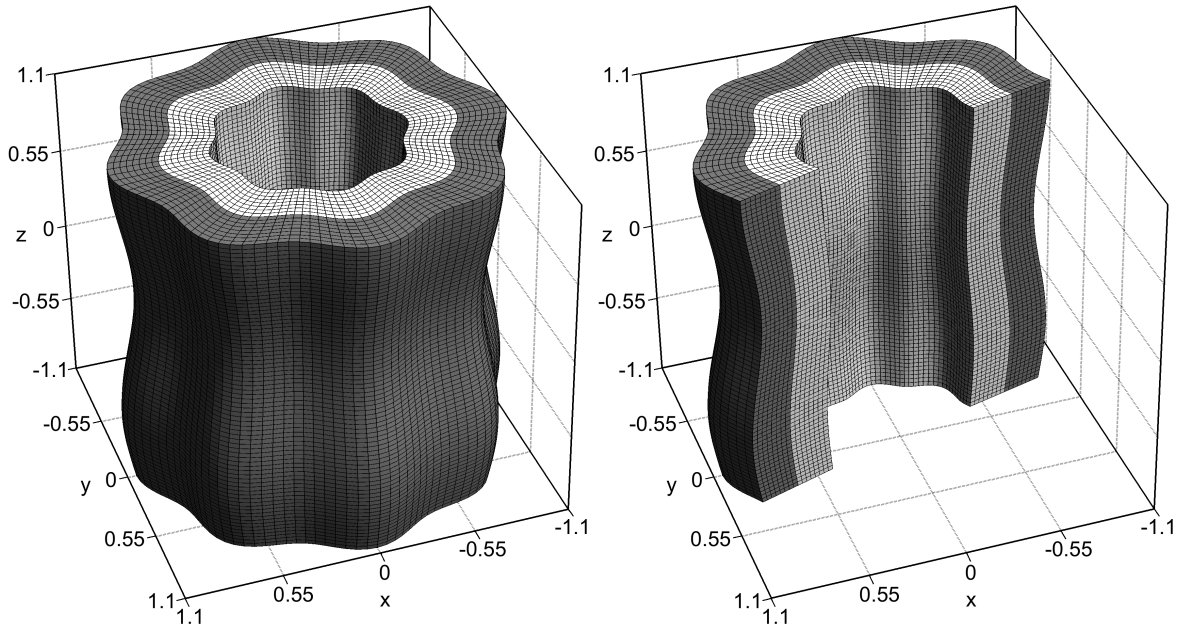
$$c = \sqrt{(T(r_*, \theta, z))^2 + (\partial_\theta T(r_*, \theta, z))^2 + (T(r_*, \theta, z)\partial_z T(r_*, \theta, z))^2}. \tag{5.132}$$

The sign in vector function $\boldsymbol{n}^\mathrm{A}(\theta, z)$ is negative on the internal surface and positive on the middle surface, whereas the sign in vector function $\boldsymbol{n}^\mathrm{B}(\theta, z)$ is positive on the external surface and negative on the middle surface. Notice that $\boldsymbol{n}^\mathrm{B} = -\boldsymbol{n}^\mathrm{A}$ on the middle surface. Radius r_* is also assigned according



(a) Full view. (b) Cross-section view.

Figure 5.9: Uniform Delaunay tetrahedral meshes for the wavy sleeve domain test case.



(a) Full view. (b) Cross-section view.

Figure 5.10: Uniform hexahedral structured meshes for the wavy sleeve domain test case.

to the surface, namely $r_* = r_{\mathrm{I}}$ for the internal surface, $r_* = r_{\mathrm{E}}$ for the external surface and $r = r_{\mathrm{M}}$ for the middle surface. Partial derivatives $\partial_\theta T(r,\theta,z) = \partial T(r,\theta,z)/\partial\theta$ and $\partial_z T(r,\theta,z) = \partial T(r,\theta,z)/\partial z$ are given as

$$\partial_\theta T(r,\theta,z) = r\Big(-\alpha_\theta\beta_\theta \sin(\alpha_\theta\theta)\Big)\Big(1+\beta_z \sin\Big(\alpha_z\Big(\frac{z-z_{\mathrm{B}}}{z_{\mathrm{T}}-z_{\mathrm{B}}}\Big)\Big)\Big), \tag{5.133}$$

$$\partial_z T(r,\theta,z) = r\Big(1+\beta_\theta \cos(\alpha_\theta\theta)\Big)\Big(\alpha_z\Big(\frac{1}{z_{\mathrm{T}}-z_{\mathrm{B}}}\Big)\beta_z \cos\Big(\alpha_z\Big(\frac{z-z_{\mathrm{B}}}{z_{\mathrm{T}}-z_{\mathrm{B}}}\Big)\Big)\Big). \tag{5.134}$$

The analytic solutions assigned to domain Ω are expressed in cylindrical coordinates and are obtained applying the same diffeomorphic transformation, $T^{-1}(r,\theta,z)$, to analytic solutions assigned to domain Ω'. More precisely, the analytic solutions are given as

$$\phi^S(r,\theta,z) = \psi^S\Big(T^{-1}(r,\theta,z),\theta,z\Big), \quad \text{in } \Omega^S, \tag{5.135}$$

for $S \in \{\mathrm{A},\mathrm{B}\}$, where functions $\psi^S(r',\theta',z')$ are defined on cylinder domain Ω' and are given as

$$\psi^{\mathrm{A}}(r',\theta',z') = \xi^{\mathrm{A}}\Big(\frac{2r'-(r_{\mathrm{M}}+r_{\mathrm{I}})}{r_{\mathrm{M}}-r_{\mathrm{I}}}\Big)\cos\Big(n_\theta^{\mathrm{A}}\theta'\Big)\sin\Big(n_z^{\mathrm{A}}\Big(\frac{z'-z_{\mathrm{B}}}{z_{\mathrm{T}}-z_{\mathrm{B}}}\Big)\Big), \quad \text{in } \Omega'^{\mathrm{A}}, \tag{5.136}$$

$$\psi^{\mathrm{B}}(r',\theta',z') = \xi^{\mathrm{B}}\Big(\frac{r'-r_{\mathrm{E}}}{r_{\mathrm{E}}-r_{\mathrm{M}}}\Big)\cos\Big(n_\theta^{\mathrm{B}}\theta'\Big)\sin\Big(n_z^{\mathrm{B}}\Big(\frac{z'-z_{\mathrm{B}}}{z_{\mathrm{T}}-z_{\mathrm{B}}}\Big)\Big), \quad \text{in } \Omega'^{\mathrm{B}}, \tag{5.137}$$

with parameters $n_\theta^{\mathrm{A}}, n_z^{\mathrm{A}}, n_\theta^{\mathrm{B}}, n_z^{\mathrm{B}} \in \mathbb{N}$, and univariate functions $\xi^S(\gamma)$ are given as

$$\xi^{\mathrm{A}}(\gamma) = a^{\mathrm{A}}\Big(\exp\Big(d^{\mathrm{A}}\gamma\Big)+\exp\Big(-d^{\mathrm{A}}\gamma\Big)+b^{\mathrm{A}}\Big), \quad \text{in } [-1,1], \tag{5.138}$$

$$\xi^{\mathrm{B}}(\gamma) = a^{\mathrm{B}}\Big(\exp\Big(d^{\mathrm{B}}\gamma\Big)+\exp\Big(-d^{\mathrm{B}}\gamma\Big)+b^{\mathrm{B}}\Big)+c^{\mathrm{B}}(\cos(\pi\gamma)+1), \quad \text{in } [-1,1], \tag{5.139}$$

with parameters $a^{\mathrm{A}}, b^{\mathrm{A}}, a^{\mathrm{B}}, b^{\mathrm{B}}, c^{\mathrm{B}}, d^{\mathrm{A}}, d^{\mathrm{B}} \in \mathbb{R}$. Notice that, although the analytic solutions are provided in cylindrical coordinates, the problem is numerically solved in Cartesian coordinates. The associated source term functions are obtained after substituting analytic solutions (5.135) into Equation (5.1).

The bottom surfaces of physical subdomains Ω^{A} and Ω^{B} correspond to physical boundary subsets $\Gamma^{\mathrm{R,A}}$ and $\Gamma^{\mathrm{R,B}}$, respectively, and both are assigned with an homogeneous Dirichlet boundary condition, for which coefficient functions $\alpha^{\mathrm{R,A}}(\boldsymbol{x}) = 1$ and $\beta^{\mathrm{R,A}}(\boldsymbol{x}) = 0$ are assigned on the former and coefficient functions $\alpha^{\mathrm{R,B}}(\boldsymbol{x}) = 1$ and $\beta^{\mathrm{R,B}}(\boldsymbol{x}) = 0$ are assigned on the latter such that the Robin boundary conditions recovers the Dirichlet boundary conditions. The top surfaces of physical subdomains Ω^{A} and Ω^{B} correspond to physical boundary subsets $\Gamma^{\mathrm{R,A}}$ and $\Gamma^{\mathrm{R,B}}$, respectively, and both are assigned with an homogeneous Neumann boundary condition (adiabatic surfaces), for which coefficient functions $\alpha^{\mathrm{R,A}}(\boldsymbol{x}) = 0$ and $\beta^{\mathrm{R,A}}(\boldsymbol{x}) = 1$ are assigned on the former and coefficient functions $\alpha^{\mathrm{R,B}}(\boldsymbol{x}) = 0$ and $\beta^{\mathrm{R,B}}(\boldsymbol{x}) = 1$ are assigned on the latter such that the Robin boundary

conditions recovers the Neumann boundary conditions. The internal surface of physical subdomain Ω^{A} and external surface of physical subdomain Ω^{B} also correspond to physical boundary subsets $\Gamma^{\mathrm{R,A}}$ and $\Gamma^{\mathrm{R,B}}$, respectively, and both are assigned with a Neumann boundary condition, with boundary condition functions given as

$$g^{\mathrm{R},S}(\theta,z) = -\kappa^{S}(r,\theta,z)\frac{\partial \psi^{S}(T(r,\theta,z),\theta,z)}{\partial r'}\nabla T^{-1}(r,\theta,z)\cdot \boldsymbol{n}^{S}(r,\theta,z), \quad \text{in } \Gamma^{\mathrm{N},S}, \tag{5.140}$$

for $S \in \{\mathrm{A},\mathrm{B}\}$, for which coefficient functions $\alpha^{\mathrm{R,A}}(\boldsymbol{x}) = 0$ and $\beta^{\mathrm{R,A}}(\boldsymbol{x}) = 1$ are assigned on the former and coefficient functions $\alpha^{\mathrm{R,B}}(\boldsymbol{x}) = 0$ and $\beta^{\mathrm{R,B}}(\boldsymbol{x}) = 1$ are assigned on the latter such that the Robin boundary conditions recovers the Neumann boundary conditions.

The middle surface between physical subdomains Ω^{A} and Ω^{B} corresponds to physical interface subset Γ^{C} where the continuity interface conditions are prescribed. Parameters $a^{\mathrm{A}}, a^{\mathrm{B}}, b^{\mathrm{A}}, b^{\mathrm{B}}, c^{\mathrm{B}}$ are determined such that the boundary conditions and the continuity interface conditions are simultaneously satisfied. Constant thermal conductivity functions $\kappa^{\mathrm{A}}(\boldsymbol{x})$ and $\kappa^{\mathrm{B}}(\boldsymbol{x})$ are assigned and parameters are determined as

$$a^{\mathrm{A}} = -\exp\left(d^{\mathrm{A}}\right)\left(\exp\left(d^{\mathrm{A}}\right)-1\right)^{-2}, \tag{5.141}$$

$$b^{\mathrm{A}} = -\exp\left(-d^{\mathrm{A}}\right)\left(\exp\left(2d^{\mathrm{A}}\right)+1\right), \tag{5.142}$$

$$\begin{aligned} a^{\mathrm{B}} = {} & \left(2\kappa^{\mathrm{A}} d^{\mathrm{A}} \exp\left(d^{\mathrm{B}}\right)\left(\exp\left(d^{\mathrm{A}}\right)+1\right)\right) \\ & \left(\kappa^{\mathrm{B}} d^{\mathrm{B}}\left(\exp\left(d^{\mathrm{A}}\right)-1\right)\left(\exp\left(2d^{\mathrm{B}}\right)-1\right)\right)^{-1}, \end{aligned} \tag{5.143}$$

$$b^{\mathrm{B}} = -\exp\left(-d^{\mathrm{B}}\right)\left(\exp\left(2d^{\mathrm{B}}\right)+1\right), \tag{5.144}$$

$$\begin{aligned} c^{\mathrm{B}} = {} & \left(\frac{\kappa^{\mathrm{A}} d^{\mathrm{A}}}{\kappa^{\mathrm{B}} d^{\mathrm{B}}}\right)\left(1+\exp\left(d^{\mathrm{A}}\right)\right) \\ & \left(1-\exp\left(d^{\mathrm{B}}\right)\right)\left(\left(1-\exp\left(d^{\mathrm{A}}\right)\right)\left(1+\exp\left(d^{\mathrm{B}}\right)\right)\right)^{-1} + \frac{1}{2}. \end{aligned} \tag{5.145}$$

Density and heat capacity functions are constant while the velocity vector fields are tangent to the surfaces such that no convection occurs through the boundaries and interface, given in cylindrical coordinates as

$$\begin{aligned} \boldsymbol{u}^{S}(r,z) = {} & \omega^{S} r\left(\alpha_z\left(\frac{1}{z_{\mathrm{T}}-z_{\mathrm{B}}}\right)\beta_z \cos\left(\alpha_z\left(\frac{z-z_{\mathrm{B}}}{z_{\mathrm{T}}-z_{\mathrm{B}}}\right)\right)\right)\left(1+\beta_z \sin\left(\alpha_z\left(\frac{z-z_{\mathrm{B}}}{z_{\mathrm{T}}-z_{\mathrm{B}}}\right)\right)\right)^{-1}\hat{\boldsymbol{r}} \\ & + \omega^{S}\hat{\boldsymbol{z}}, \quad \text{in } \Omega^{S}, \end{aligned} \tag{5.146}$$

for $S \in \{\mathrm{A},\mathrm{B}\}$, with parameters $\omega^{\mathrm{A}}, \omega^{\mathrm{B}} \in \mathbb{R}$.

The test case is addressed with parameters $\alpha_\theta = 8$, $\beta_\theta = 0.05$, $\alpha_z = 5\pi/2$, $\beta_z = 0.05$,

$n_\theta^{\mathrm{A}} = n_\theta^{\mathrm{B}} = 5,\ n_z^{\mathrm{A}} = n_z^{\mathrm{B}} = 7\pi/2,\ d^{\mathrm{A}} = d^{\mathrm{B}} = 1.$ The analytic solutions only depend on these parameters and on the thermal conductivity functions (through parameters a^{A}, b^{A}, a^{B}, and b^{B}), which are assigned according to the following cases:

- Low thermal conductivity ratio case: $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 1$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 2$ (see Figure 5.11).
- High thermal conductivity ratio case: $\kappa^{\mathrm{A}}(\boldsymbol{x}) = 1$ and $\kappa^{\mathrm{B}}(\boldsymbol{x}) = 100$ (see Figure 5.12),

where the ratio is defined as $\kappa^{\mathrm{B}}(\boldsymbol{x})/\kappa^{\mathrm{A}}(\boldsymbol{x})$. Parameters ω^{A} and ω^{B} in the velocity functions are assigned according to the following cases:

- Low Péclet number ratio case: $\omega^{\mathrm{A}} = 1$ and $\omega^{\mathrm{B}} = 2$.
- High Péclet number ratio case: $\omega^{\mathrm{A}} = 100$ and $\omega^{\mathrm{B}} = 1$,

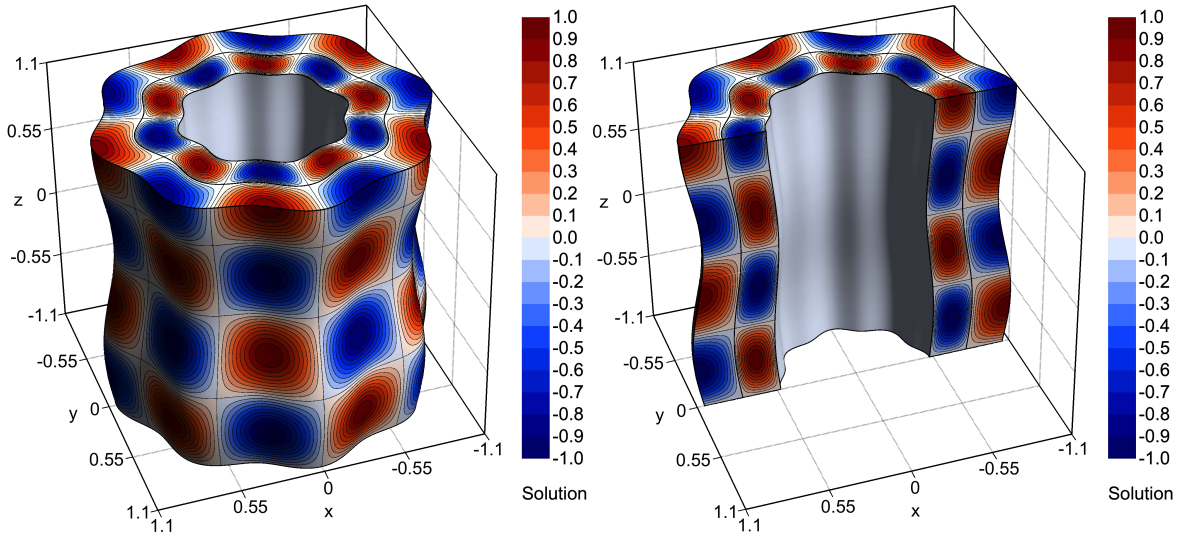
where the ratio is defined as $\mathrm{Pe}^{\mathrm{B}}/\mathrm{Pe}^{\mathrm{A}}$, with $\mathrm{Pe}^S = \omega^S/\kappa^S$, $S \in \{\mathrm{A}, \mathrm{B}\}$, the Péclet number in physical subdomain Ω^S (the characteristic dimension is unitary).

Successively finer uniform Delaunay tetrahedral meshes or uniform structured hexahedral meshes (with no curved elements) are used to discretize each physical subdomain (see Figure 5.9 and 5.10) and the simulations are carried out for configurations DN and ND and for polynomial degrees $d = 1, 3, 5$.

5.5.2.1 Low Péclet number ratio case

The low Péclet number ratio case combined with the low and high thermal conductivity ratio cases are addressed, and simulations are carried out with successive finer uniform Delaunay tetrahedral meshes, polynomial degrees $d = 1, 3, 5$, and configurations DN and ND. The corresponding Péclet numbers are $\mathrm{Pe}^{\mathrm{A}} = 1$ and $\mathrm{Pe}^{\mathrm{B}} = 1$ for the former combination and $\mathrm{Pe}^{\mathrm{A}} = 1$ and $\mathrm{Pe}^{\mathrm{B}} = 0.02$ for the latter combination. The relative errors and convergence orders are reported in Table 5.10. Notice that configuration DN prescribes the Neumann boundary condition in subproblem P^{A} (with the highest thermal conductivity), while the Dirichlet boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity). Again, configuration ND prescribes the opposite situation.

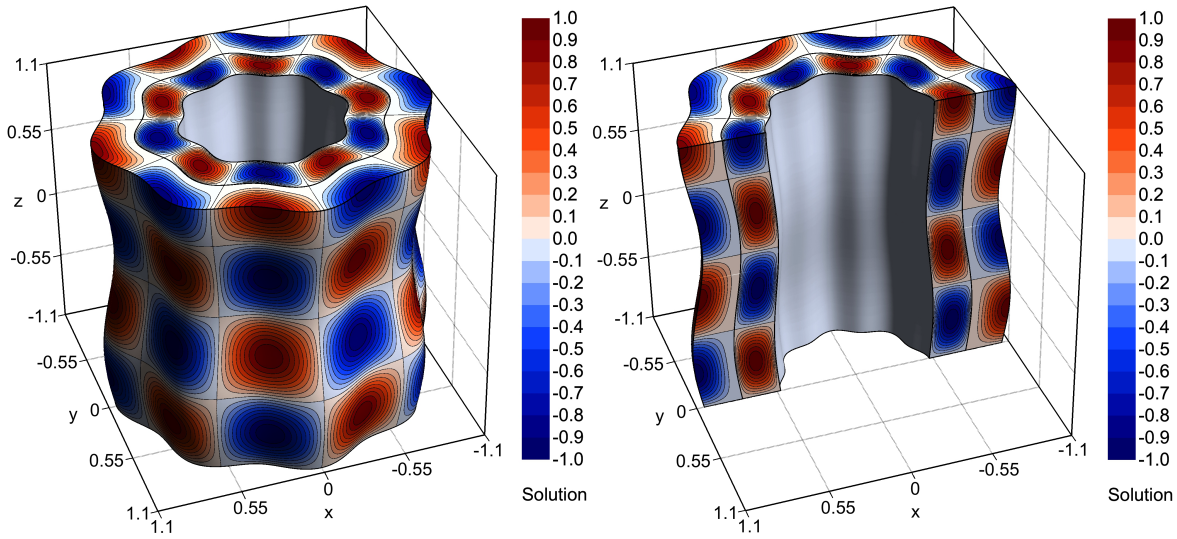
In both thermal conductivity ratio cases and regardless of the partitioning configuration, the second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms. Following the previous cases, the accuracy provided from configurations DN and ND is comparable regardless of the polynomial degree, where the difference is exceptionally negligible for lower thermal conductivity ratios, that is, higher thermal interaction. Additionally, the impact of the thermal conductivity ratio on the accuracy is also low.



(a) Full view.

(b) Cross-section view.

Figure 5.11: Analytic solution for the wavy sleeve domain test case with a low thermal conductivity ratio.



(a) Full view.

(b) Cross-section view.

Figure 5.12: Analytic solution for the wavy sleeve domain test case with a high thermal conductivity ratio.

The number of GMRES iterations are reported in Table 5.11. In both thermal conductivity ratio cases, the number of iterations obtained with both partitioning configurations is comparable using polynomial degrees $d = 1, 3$. With polynomial degree $d = 5$ and in the low thermal conductivity ratio case, a considerably higher number of iterations is required with configuration ND than with configuration DN. The results differ from the previous test case, where the number of iterations is comparable with both partitioning configurations and any polynomial degree. On the other side, in the high thermal conductivity ratio case, a larger number of iterations are required with configuration ND than with configuration DN and, indeed, the former becomes computationally very demanding. In general, more iterations are required in the low thermal conductivity ratio case than with the high thermal conductivity ratio case for the same partitioning configuration and polynomial degree. Indeed, such behaviour is expected since, in the former case, the thermal interaction between physical subdomains is stronger.

Similar to the previous case, the same test case is addressed with successive finer uniform structured hexahedral meshes. The relative errors and convergence orders are reported in Table 5.12.

In both thermal conductivity ratio cases, the second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms, with configuration DN. Once again, configuration ND provides similar behaviour as before with uniform Delaunay tetrahedral meshes and is not reported for the sake of compactness. Notice that no specific treatment is given according to the type of mesh elements. In that way, besides its capability to provide very high-order of convergence, the results also support the versatility and simplicity of the proposed method to handle general polyhedral meshes generated for curved domains.

5.5.2.2 High Péclet number ratio case

The high Péclet number ratio case combined with the low and high thermal conductivity ratio cases are addressed, and simulations are carried out with successive finer uniform Delaunay tetrahedral meshes, polynomial degrees $d = 1, 3, 5$, and configurations DN and ND. The corresponding Péclet numbers are $\mathrm{Pe}^{\mathrm{A}} = 100$ and $\mathrm{Pe}^{\mathrm{B}} = 0.5$ for the former combination and $\mathrm{Pe}^{\mathrm{A}} = 100$ and $\mathrm{Pe}^{\mathrm{B}} = 0.01$ for the latter combination. The relative errors and convergence orders are reported in Table 5.13. Notice that configuration DN prescribes the Neumann boundary condition in subproblem P^{A} (with the highest thermal conductivity), while the Dirichlet boundary condition is prescribed in subproblem P^{B} (with the lowest thermal conductivity). Again, configuration ND prescribes the opposite situation. In many practical problems, the interface consists of a fluid in contact with a solid exchanging heat and, therefore, the two physical subdomains have very distinct Péclet numbers, as this case aims at replicating.

The second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms. Form the accuracy viewpoint, both partitioning configurations provide similar accuracy, and the influence of the thermal conductivity

Table 5.10: Relative errors and convergence orders obtained in the wavy sleeve domain test case with a low Péclet number ratio and uniform Delaunay tetrahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio, configuration DN												
189 280	8.32E−03	—	5.40E−02	—	2.39E−04	—	3.78E−03	—	3.84E−05	—	2.39E−03	—
534 544	3.85E−03	2.23	2.72E−02	1.98	5.44E−05	4.28	8.08E−04	4.46	4.08E−06	6.48	2.08E−04	7.05
1 642 994	1.74E−03	2.12	1.40E−02	1.78	1.15E−05	4.16	2.85E−04	2.78	4.00E−07	6.20	2.07E−05	6.17
5 248 131	7.76E−04	2.08	7.27E−03	1.68	2.33E−06	4.12	3.32E−05	5.55	3.91E−08	6.01	2.25E−06	5.73
Low thermal conductivity ratio, configuration ND												
189 280	7.24E−03	—	4.69E−02	—	2.22E−04	—	3.72E−03	—	3.71E−05	—	2.38E−03	—
534 544	3.32E−03	2.25	2.53E−02	1.78	5.01E−05	4.31	6.79E−04	4.91	3.94E−06	6.48	1.81E−04	7.45
1 642 994	1.49E−03	2.14	1.12E−02	2.19	1.03E−05	4.23	2.85E−04	2.32	3.77E−07	6.27	1.71E−05	6.30
5 248 131	6.63E−04	2.09	5.80E−03	1.69	2.07E−06	4.14	3.22E−05	5.63	3.52E−08	6.12	2.07E−06	5.46
High thermal conductivity ratio, configuration DN												
189 280	9.35E−03	—	7.61E−02	—	3.16E−04	—	3.50E−03	—	5.36E−05	—	2.36E−03	—
534 544	4.50E−03	2.11	3.83E−02	1.99	7.21E−05	4.27	9.11E−04	3.89	6.05E−06	6.30	1.87E−04	7.32
1 642 994	2.08E−03	2.06	2.10E−02	1.60	1.52E−05	4.15	2.84E−04	3.11	6.17E−07	6.10	1.51E−05	6.73
5 248 131	9.36E−04	2.06	9.12E−03	2.16	3.12E−06	4.10	1.15E−04	2.34	7.34E−08	5.50	2.89E−06	4.28
High thermal conductivity ratio, configuration ND												
189 280	7.72E−03	—	7.61E−02	—	3.18E−04	—	3.50E−03	—	5.52E−05	—	2.36E−03	—
534 544	3.67E−03	2.15	3.82E−02	1.99	7.27E−05	4.26	9.10E−04	3.89	6.31E−06	6.27	1.86E−04	7.34
1 642 994	1.68E−03	2.08	2.10E−02	1.60	1.53E−05	4.17	2.86E−04	3.09	6.26E−07	6.17	1.52E−05	6.68
5 248 131	7.54E−04	2.08	9.12E−03	2.15	3.15E−06	4.08	1.15E−04	2.36	6.13E−08	6.00	2.30E−06	4.89

Table 5.11: Number of GMRES iterations obtained in the wavy sleeve domain test case with a low Péclet number ratio and uniform Delaunay tetrahedral meshes.

	Low thermal conductivity ratio						High thermal conductivity ratio					
	Configuration DN			Configuration ND			Configuration DN			Configuration ND		
DOF	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$
189 280	48	120	306	55	132	225	37	82	119	49	114	411
534 544	71	186	391	78	207	374	52	125	179	62	171	7 229
1 642 994	106	280	487	117	319	794	83	201	241	104	275	>200 000
5 248 131	167	436	628	182	491	1 451	131	318	338	157	432	>200 000

Table 5.12: Relative errors and convergence orders obtained in the wavy sleeve domain test case with a low Péclet number ratio, configuration DN, and uniform structured hexahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio												
141 056	6.99E−03	–	4.17E−02	–	6.50E−04	–	1.11E−02	–	3.38E−04	–	7.08E−03	–
417 216	4.24E−03	1.38	3.02E−02	0.89	1.50E−04	4.05	2.90E−03	3.71	4.25E−05	5.74	1.04E−03	5.32
1 291 392	2.66E−03	1.24	2.10E−02	0.96	3.36E−05	3.98	6.69E−04	3.89	4.02E−06	6.26	1.29E−04	5.52
4 421 520	1.68E−03	1.12	1.45E−02	0.90	6.69E−06	3.93	1.42E−04	3.78	3.29E−07	6.10	1.34E−05	5.53
High thermal conductivity ratio												
141 056	9.90E−03	–	5.52E−02	–	6.55E−04	–	1.04E−02	–	3.04E−04	–	7.03E−03	–
417 216	5.57E−03	1.59	3.50E−02	1.26	1.65E−04	3.81	2.72E−03	3.70	4.01E−05	5.60	1.01E−03	5.36
1 291 392	3.25E−03	1.43	2.34E−02	1.07	3.77E−05	3.92	6.87E−04	3.65	4.34E−06	5.90	1.30E−04	5.46
4 421 520	1.94E−03	1.26	1.59E−02	0.94	7.81E−06	3.84	1.42E−04	3.84	3.89E−07	5.88	1.36E−05	5.50

ratio is also low. Moreover, the approximate solutions have essentially non-oscillatory behaviour, and the higher Péclet number ratio case is effectively handled without deterioration of accuracy or convergence order.

The number of GMRES iterations are reported in Tables 5.14 and shows that the importance of choosing the appropriate partitioning configuration becomes unequivocal. Indeed, in the high thermal conductivity ratio case with configuration ND and polynomial degree $d = 5$, the residual convergence in the GMRES method attains the tolerance parameter within a reasonable, but considerably higher, number of iterations with the first mesh. In contrast, with the other meshes, it is crucial to choose configuration DN to achieve convergence. Nevertheless, although the differences are not so prominent in the remaining case and polynomial degrees, configuration DN still performs more efficiently than configuration ND. Therefore, prescribing the Dirichlet boundary condition resulting from the partitioning method in the subproblem with the lower thermal conductivity (and the Neumann boundary condition in the other) becomes particularly crucial for higher Péclet number ratios and higher thermal conductivity ratios.

To conclude the numerical benchmark, the same test case is addressed with successive finer uniform structured hexahedral meshes and the relative errors and convergence orders are reported in Table 5.15.

The second-, fourth-, and sixth-orders of convergence are effectively achieved with polynomial degrees $d = 1, 3, 5$, respectively, both in terms of the L^1- and L^∞-norms, with configurations DN. Configuration ND leads to the same conclusions but is not reported for the sake of compactness.

The results for the wavy sleeve domain test case strongly support the capability, robustness, and stability of the proposed method to handle non-trivial three-dimensional domains with curved boundaries and interfaces and provide very high-order of convergence. The classical techniques based

Table 5.13: Relative errors and convergence orders obtained in the wavy sleeve domain test case with a high Péclet number ratio and uniform Delaunay tetrahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio, configuration DN												
189 280	8.32E−03	—	5.40E−02	—	2.39E−04	—	3.78E−03	—	3.84E−05	—	2.39E−03	—
534 544	3.85E−03	2.23	2.72E−02	1.98	5.44E−05	4.28	8.08E−04	4.46	4.08E−06	6.48	2.08E−04	7.05
1 642 994	1.74E−03	2.12	1.40E−02	1.78	1.15E−05	4.16	2.85E−04	2.78	4.00E−07	6.20	2.07E−05	6.17
5 248 131	7.76E−04	2.08	7.27E−03	1.68	2.33E−06	4.12	3.32E−05	5.55	3.91E−08	6.01	2.25E−06	5.73
Low thermal conductivity ratio, configuration ND												
189 280	7.24E−03	—	4.69E−02	—	2.22E−04	—	3.72E−03	—	3.71E−05	—	2.38E−03	—
534 544	3.32E−03	2.25	2.53E−02	1.78	5.01E−05	4.31	6.79E−04	4.91	3.94E−06	6.48	1.81E−04	7.45
1 642 994	1.49E−03	2.14	1.12E−02	2.19	1.03E−05	4.23	2.85E−04	2.32	3.77E−07	6.27	1.71E−05	6.30
5 248 131	6.63E−04	2.09	5.80E−03	1.69	2.07E−06	4.14	3.22E−05	5.63	3.52E−08	6.12	2.07E−06	5.46
High thermal conductivity ratio, configuration DN												
189 280	9.35E−03	—	7.61E−02	—	3.16E−04	—	3.50E−03	—	5.36E−05	—	2.36E−03	—
534 544	4.50E−03	2.11	3.83E−02	1.99	7.21E−05	4.27	9.11E−04	3.89	6.05E−06	6.30	1.87E−04	7.32
1 642 994	2.08E−03	2.06	2.10E−02	1.60	1.52E−05	4.15	2.84E−04	3.11	6.17E−07	6.10	1.51E−05	6.73
5 248 131	9.36E−04	2.06	9.12E−03	2.16	3.12E−06	4.10	1.15E−04	2.34	7.34E−08	5.50	2.89E−06	4.28
High thermal conductivity ratio, configuration ND												
189 280	7.72E−03	—	7.61E−02	—	3.18E−04	—	3.50E−03	—	5.52E−05	—	2.36E−03	—
534 544	3.67E−03	2.15	3.82E−02	1.99	7.27E−05	4.26	9.10E−04	3.89	6.31E−06	6.27	1.86E−04	7.34
1 642 994	1.68E−03	2.08	2.10E−02	1.60	1.53E−05	4.17	2.86E−04	3.09	6.26E−07	6.17	1.52E−05	6.68
5 248 131	7.54E−04	2.08	9.12E−03	2.15	3.15E−06	4.08	1.15E−04	2.36	6.13E−08	6.00	2.30E−06	4.89

Table 5.14: Number of GMRES iterations obtained in the wavy sleeve domain test case with a high Péclet number ratio and uniform Delaunay tetrahedral meshes.

DOF	Low thermal conductivity ratio						High thermal conductivity ratio					
	Configuration DN			Configuration ND			Configuration DN			Configuration ND		
	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$	$d=1$	$d=3$	$d=5$
189 280	91	124	251	92	131	169	85	118	143	92	143	481
534 544	129	187	325	131	197	297	126	172	216	131	215	>200 000
1 642 994	186	290	488	190	311	496	185	276	304	195	340	>200 000
5 248 131	286	454	672	300	493	929	272	432	459	302	600	>200 000

Table 5.15: Relative errors and convergence orders obtained in the wavy sleeve domain test case with a high Péclet number ratio, configuration DN, and uniform structured hexahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Low thermal conductivity ratio												
141 056	5.46E−03	–	3.08E−02	–	4.40E−04	–	5.71E−03	–	2.34E−04	–	4.99E−03	–
417 216	3.15E−03	1.52	1.90E−02	1.34	1.04E−04	3.99	1.62E−03	3.49	2.72E−05	5.95	6.73E−04	5.54
1 291 392	1.89E−03	1.36	1.16E−02	1.31	2.41E−05	3.88	4.10E−04	3.65	2.67E−06	6.16	7.46E−05	5.84
4 421 520	1.15E−03	1.22	7.34E−03	1.11	4.86E−06	3.90	8.93E−05	3.72	2.17E−07	6.13	6.91E−06	5.80
High thermal conductivity ratio												
141 056	7.01E−03	–	3.70E−02	–	4.47E−04	–	5.70E−03	–	2.17E−04	–	5.01E−03	–
417 216	3.85E−03	1.66	1.93E−02	1.80	1.15E−04	3.74	1.62E−03	3.48	2.48E−05	6.00	6.67E−04	5.58
1 291 392	2.19E−03	1.49	1.06E−02	1.60	2.73E−05	3.83	4.10E−04	3.65	2.65E−06	5.94	7.46E−05	5.82
4 421 520	1.28E−03	1.31	7.27E−03	0.91	5.73E−06	3.80	8.93E−05	3.72	2.38E−07	5.88	6.91E−06	5.80

on curved mesh elements to preserve the optimal convergence order would be cumbersome and a challenging task to apply in such a domain due to the complexity of the diffeomorphic transformation. In that case, it turns out very complex to perform the non-linear transformations (in the case of the isoparametric elements) and to determine the appropriate quadrature rules. On the other side, the proposed method cleverly preserves the optimal convergence order using, for the purpose, collocation points located on the curved physical boundaries and interface. Besides its simplicity, the proposed method performs efficiently in a wide range of thermal situations for several Péclet numbers and thermal conductivity.

5.6 Computational benchmark

The assessment of the computational efficiency of the proposed method is based on the performance analysis in terms of execution time and memory usage to solve conjugate heat transfer problems in arbitrary three-dimensional curved domains. The four steps that comprise a simple implementation of the proposed method in an implicit formulation are gradually optimized to reduce the computational cost and improve efficiency. The computational gain of each optimization is assessed with the following algorithms:

- Algorithm 1: step 1 is optimized (see Section 5.4.1).
- Algorithm 2: step 2 is optimized and step 1 is eliminated from the implementation (see Section 5.4.2).
- Algorithm 3: step 3 is optimized and steps 1 and 2 are eliminated from the implementation (see Section 5.4.3).

- Algorithm 4: step 4 is optimized and steps 1, 2, and 3 are eliminated from the implementation (see Section 5.4.4).

Algorithms 1, 2, and 3 provide an implicit formulation of the proposed method through a global residual operator, whereas algorithm 4 provides an explicit formulation through a coefficients matrix and a right-hand side vector. In both cases, an iterative method is supplied to solve the implicit or explicit system of linear equations. The algorithms are implemented in an in-house library written in Fortran with multiprocessing capabilities to take advantage of high-performance computing platforms.

The same problem is solved regardless of the algorithm since only the implementation is modified deriving from the algebraic rearrangement of the proposed method without introducing more approximations. Hence, the following parameters are reported:

- Residual time, denoted as T_{R} (in seconds (s)), corresponds to the execution time to compute the global residual operator (in the implicit formulation) or the matrix-vector product (in the explicit formulation). The relative residual time, denoted as R_{R} (in percentage (%)), is determined taking the residual time of algorithm 1 for the same mesh as the reference.

- Solution time, denoted as T_{S} (in seconds (s)), corresponds to the execution time to solve the implicit or explicit system of linear equations. The relative solution time, denoted as R_{S} (in percentage (%)) is determined taking the solution time of algorithm 1 for the same mesh as the reference.

- Memory usage, denoted as M (in gigabytes (Gb)), corresponds to the maximum random access memory allocated for the simulation (virtual memory high watermark). The relative memory usage, denoted as R_{M} (in percentage (%)) is determined taking the memory usage of algorithm 1 for the same mesh as the reference.

The GMRES method is used for both the implicit and explicit formulations, but other methods for non-symmetric systems of linear equations can be applied, such as the BiCGSTAB method. However, they are not addressed for the sake of compactness. The GMRES method comprises, not only computing the global residual operator (in the implicit formulation) or the matrix-vector product (in the explicit formulation) at each iteration but also constructing the Krylov basis through Arnoldi iterations and checking the stopping criteria. The solution time corresponds to the time elapsed from the first iteration to the last, whereas the residual time corresponds only to time spent on the proposed method and, hence, $T_{\mathrm{S}} > T_{\mathrm{R}}$. The solution time increases quadratically with the number of iterations due to the Krylov basis (assuming that the method is not restarted) and also depends on the iterative method applied. On the other side, the residual time increases linearly since the computational cost to compute the residual operator or the matrix-vector product at each iteration does not depend on the number of iterations. Similarly, the memory usage also increases linearly since each additional iteration requires one additional vector of constant size to store for the Krylov basis. In that regard, the

computational gains of the algorithms in terms of relative residual time and relative memory usage do not depend on the number of iterations. Contrarily, the computational gains in terms of solution time strongly depend on the problem (as more or fewer iterations are needed) and, therefore, the analysis is performed for a fixed number of 200 iterations. Moreover, keeping same number of iterations, it is possible to better perceive the relative computational cost of a single iteration according to the polynomial degree. The advantage of reporting the relative residual time, in addition to the relative solution time, is that the former depends neither on the number of iterations nor on the iterative method. Therefore, the relative residual time assesses more precisely the effective computational gain of the algorithms (since the iterative method is not part of the optimization).

Besides the algorithm and the number of iterations required to solve the implicit or explicit system of linear equations, the computational cost of the proposed method also depends on the mesh and the polynomial degree. On the other side, the problem characteristics (geometry, physical properties, boundary and interface conditions, among others) do not have a direct impact on the computational cost. Indeed, they do not change the number of floating-point operations, either in the global residual operator or in the matrix-vector product, but rather the numerical result. Therefore, a standard test case is considered, and the analysis is performed for algorithms 1 to 4, successively finer meshes, and polynomial degrees $d = 1, 3, 5$. Concerning the mesh, not only the number of cells has an impact on the computational cost, but also the type of elements since it determines the ratio between faces and cells and the associated number of quadrature points. Therefore, the benchmark is split into tetrahedral, prismatic, and hexahedral meshes.

The quadrature rules for the elements composing the prismatic meshes consist of splitting the quadrilateral faces into two triangular sections and the cells into three tetrahedral sections. Similarly, for the case of hexahedral meshes, each quadrilateral face is split into two triangular sections and each cell is split into five tetrahedral sections. Then, the standard quadrature rules for triangles are applied in each triangular face, and each triangular portion of the quadrilateral faces and, similarly, the standard quadrature rules for tetrahedrons are used in each tetrahedral portion of the prismatic cells. Another strategy consists in deriving specific quadrature rules for prismatic elements, often more efficient in terms of the number of quadrature points to provide the same convergence order for the integration error, which are then applied straightforwardly. However, the splitting approach turns out to be more general and can be applied straightforwardly for any kind of mesh element, including eventually non-coplanar faces. Moreover, the proposed algorithms should ultimately lead to a similar computational cost regardless of the number of quadrature points, which becomes another fundamental reason for an optimized implementation.

5.6.1 Tetrahedral meshes

Simulations with successive finer tetrahedral meshes, polynomial degrees $d = 1, 3, 5$, and algorithms 1 to 4 are carried out. The residual time, solution time, and corresponding relative

values are reported in Table 5.16. Notice that, with algorithm 1 and polynomial degree $d = 5$, the residual time and solution time are not provided for the last mesh since the amount of memory for the simulation exceeds the available memory. Hence, the relative residual time and solution time concerning algorithms 2, 3, and 4 can not be determined.

Regardless of the polynomial degree and the number of degrees of freedom, the computational gain of the simulations increases from algorithm 1 to 4, either in terms of residual time or solution time. Moreover, the solution time is always higher than the residual time, and the same is observed concerning the relative values. Nevertheless, the relative residual time and solution time are not constant with the number of degrees of freedom, possibly due to cache and memory effects that are difficult to control. To sum up, the following computational gains in terms of execution time are obtained for the finest tetrahedral mesh and depicted in Figure 5.13:

- For polynomial degree $d = 1$, algorithms 2, 3, and 4 provide a relative residual time of 57%, 42%, and 7%, respectively, and a relative solution time of 61%, 48%, and 17%, respectively.
- For polynomial degree $d = 3$, algorithms 2, 3, and 4 provide a relative residual time of 34%, 16%, and 4%, respectively, and a relative solution time of 35%, 18%, and 7%, respectively.
- For polynomial degree $d = 5$, algorithms 2, 3, and 4 provide a relative residual time of 30%, 7%, and 2%, respectively, and a relative solution time of 30%, 8%, and 3%, respectively.

For the second-order of convergence, with polynomial degree $d = 1$, the implementation is significantly optimized with algorithm 4 providing a relative solution time of 17% (factor of 6). As expected, for a fourth- and sixth-orders of convergence, with polynomial degrees $d = 3, 5$, respectively, the computational gains are more significant with algorithm 4 providing relative solution time of 7% (14 times faster) and 3% (33 times faster), respectively. The results prove that the implementation of the proposed method in the explicit formulation is computationally more efficient than any optimized implementation in the implicit formulation, and a significant computational gain is provided without any accuracy or convergence order deterioration. Moreover, with algorithm 4 and polynomial degrees $d = 1, 3, 5$, the relative solution time is approximately 5, 3, and 1.5 times higher, respectively, than the corresponding relative residual time. Indeed, when the polynomial degree increases, the computational cost of the implementation of the explicit formulation increases since the coefficients matrix becomes denser and, therefore, the residual time approaches the solution time, and the difference becomes smaller.

The memory usage and corresponding relative values are reported in Table 5.17. Notice that, with algorithm 1 and polynomial degree $d = 5$, the memory usage is not provided for the last mesh since the amount requires for the simulations exceeds the available resources. Hence, the relative memory usage concerning algorithms 2, 3, and 4 can not be determined.

The memory usage is proportional to the number of degrees of freedom and increases as the polynomial degree increases (due to larger coefficients vectors, stencils, least-squares matrices).

Table 5.16: Residual time, solution time, and relative values obtained with tetrahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	T_R [s]	R_R [%]	T_S [s]	R_S [%]	T_R [s]	R_R [%]	T_S [s]	R_S [%]	T_R [s]	R_R [%]	T_S [s]	R_S [%]
Algorithm 1												
80 829	2.79	100.00	3.79	100.00	15.55	100.00	16.61	100.00	70.64	100.00	71.67	100.00
254 516	9.97	100.00	11.71	100.00	46.19	100.00	48.06	100.00	216.43	100.00	218.18	100.00
726 929	29.02	100.00	32.84	100.00	141.87	100.00	146.04	100.00	595.87	100.00	599.90	100.00
2 290 543	101.53	100.00	113.84	100.00	474.46	100.00	486.36	100.00	–	–	–	–
Algorithm 2												
80 829	1.90	68.24	2.82	74.37	6.35	40.84	7.37	44.36	20.26	28.68	21.27	29.68
254 516	6.03	60.49	7.81	66.70	21.31	46.14	23.22	48.30	62.68	28.96	64.50	29.56
726 929	17.30	59.60	21.20	64.55	57.78	40.72	62.11	42.53	176.28	29.58	180.31	30.06
2 290 543	58.30	57.42	69.28	60.85	160.92	33.92	172.63	35.49	538.24	–	549.53	–
Algorithm 3												
80 829	1.66	59.68	2.54	67.03	2.67	17.17	3.49	21.03	4.63	6.56	5.55	7.74
254 516	4.55	45.67	6.23	53.18	7.83	16.96	9.64	20.06	14.78	6.83	16.51	7.57
726 929	13.73	47.32	17.65	53.75	24.56	17.31	28.81	19.73	41.38	6.94	45.36	7.56
2 290 543	43.09	42.44	54.28	47.68	74.52	15.71	85.75	17.63	132.69	–	144.11	–
Algorithm 4												
80 829	0.20	7.10	1.07	28.30	0.68	4.34	1.64	9.90	1.54	2.18	2.50	3.49
254 516	0.76	7.59	2.41	20.60	2.26	4.89	4.01	8.35	5.07	2.34	7.01	3.21
726 929	2.17	7.48	6.07	18.49	6.42	4.52	10.38	7.11	14.74	2.47	18.84	3.14
2 290 543	6.71	6.61	19.76	17.36	20.83	4.39	32.00	6.58	47.12	–	58.51	–

Moreover, regardless of the polynomial degree, there is a reduction of memory usage from algorithms 1 to 3, whereas algorithm 4 provides higher relative memory usage than algorithm 3. Nevertheless, for a fourth- and sixth-order of convergence, with polynomial degrees $d = 3, 5$, respectively, algorithm 4 always uses considerably less memory than algorithms 1 and 2. To sum up, the following computational gains in terms of memory usage are obtained for the finest tetrahedral mesh and depicted in Figure 5.13:

- For polynomial degree $d = 1$, algorithms 2, 3, and 4 provide a relative memory usage of 93%, 87%, and 91%, respectively.
- For polynomial degree $d = 3$, algorithms 2, 3, and 4 provide a relative memory usage of 62%, 28%, and 37%, respectively.
- For polynomial degree $d = 5$, algorithms 2, 3, and 4 provide a relative memory usage of 37%, 8%, and 11%, respectively.

The relative memory usage for polynomial degree $d = 1$ does not significantly change from one

algorithm to the other. On the contrary, a significant reduction is observed with polynomial degrees $d = 3, 5$. Indeed, the implementation of the proposed method in the explicit formulation requires only 37% (factor of 3) and 11% (factor of 9) of the memory required with algorithm 1, which corresponds to a drastic reduction in terms of computational resources requirements and is significant in practice.

Table 5.17: Memory usage and relative values obtained with tetrahedral meshes.

	$d = 1$		$d = 3$		$d = 5$	
DOF	M [Gb]	R_{M} [%]	M [Gb]	R_{M} [%]	M [Gb]	R_{M} [%]
Algorithm 1						
80 829	0.91	100.00	4.00	100.00	23.78	100.00
254 516	2.65	100.00	12.15	100.00	73.01	100.00
726 929	7.43	100.00	33.95	100.00	204.70	100.00
2 290 543	22.94	100.00	104.82	100.00	–	–
Algorithm 2						
80 829	0.79	87.69	2.49	62.29	8.70	36.60
254 516	2.46	92.87	7.49	61.68	26.73	36.61
726 929	6.89	92.67	20.96	61.75	74.91	36.59
2 290 543	21.22	92.51	64.70	61.72	232.11	–
Algorithm 3						
80 829	0.80	88.27	1.11	27.70	1.88	7.90
254 516	2.38	89.59	3.45	28.40	5.71	7.83
726 929	6.47	87.01	9.46	27.87	15.86	7.75
2 290 543	19.94	86.91	29.29	27.94	49.06	–
Algorithm 4						
80 829	0.83	91.71	1.48	36.88	2.52	10.60
254 516	2.45	92.16	4.43	36.46	7.82	10.71
726 929	6.80	91.54	12.38	36.47	21.96	10.73
2 290 543	20.96	91.36	38.37	36.60	68.14	–

5.6.2 Prismatic meshes

Simulations with successive finer prismatic meshes, polynomial degrees $d = 1, 3, 5$, and algorithms 1 to 4 are carried out. The residual time, solution time, and corresponding relative values are reported in Table 5.18. Notice that, with algorithm 1 and polynomial degree $d = 5$, the residual time and solution time are not provided for the last two meshes since the amount of memory for the simulation exceeds the available resources. Hence, the relative residual time and solution time

concerning algorithms 2, 3, and 4 can not be determined.

The results for the residual time and solution time with prismatic meshes follow the observations with tetrahedral meshes. To sum up, the following computational gains in terms of execution time are obtained for the finest prismatic mesh and depicted in Figure 5.13:

- For polynomial degree $d = 1$, algorithms 2, 3, and 4 provide a relative residual time of 64%, 44%, and 6%, respectively, and a relative solution time of 67%, 48%, and 14%, respectively.
- For polynomial degree $d = 3$, algorithms 2, 3, and 4 provide a relative residual time of 51%, 15%, and 3%, respectively, and a relative solution time of 52%, 16%, and 5%, respectively.
- For polynomial degree $d = 5$, algorithms 2, 3, and 4 provide a relative residual time of 53%, 7%, and 1%, respectively, and a relative solution time of 53%, 7%, and 2%, respectively.

The second-order of convergence, with polynomial degree $d = 1$ achieves a substantial computational gain with algorithm 4 with relative solution time of 14% (7 times faster). Even more pronounced, the fourth- and sixth-orders of convergence, with polynomial degrees $d = 3, 5$, respectively, achieve a relative solution time of 5% (20 times faster) and 2% (50 times faster), respectively. Drawing a comparison with the case of tetrahedral meshes, for polynomial degree $d = 1$, the relative residual time and solution time are always higher with prismatic meshes. On the other side, for polynomial degrees $d = 3, 5$, the relative residual time and solution time are higher only with algorithm 2. In contrast, algorithms 3 and 4 exceed the computational gain obtained in the case of tetrahedral meshes. Around 60% of the total number of faces in prismatic meshes are quadrilaterals and, therefore, require twice the quadrature points of triangular faces to obtain the same convergence order of the numerical integration. Indeed, since more quadrature points are required in the quadrilateral faces of the prismatic mesh elements to obtain the same convergence order, a more significant computational gain is expected. Also worth to mention, the absolute residual time and solution time obtained with algorithm 4 for prismatic meshes are comparable with the ones obtained with the same algorithm for tetrahedral meshes. Indeed, this further emphasizes the importance of an optimized implementation, where the computational cost of the simulations does not depend on the kind of mesh elements or the number of quadrature points for each element.

The memory usage and corresponding relative values are reported in Table 5.19. Notice that, with algorithm 1 and polynomial degree $d = 5$, the memory usage is not provided for the last two meshes since the amount requires for the simulation exceeds the available resources. Hence, the relative memory usage concerning algorithms 2, 3, and 4 can not be determined.

To sum up, the following computational gains in terms of memory usage are obtained for the finest prismatic mesh and depicted in Figure 5.13:

- For polynomial degree $d = 1$, algorithms 2, 3, and 4 provide a relative memory usage of 96%, 83%, and 90%, respectively.

Table 5.18: Residual time, solution time, and relative values obtained with prismatic meshes.

	$d=1$				$d=3$				$d=5$			
DOF	T_R [s]	R_R [%]	T_S [s]	R_S [%]	T_R [s]	R_R [%]	T_S [s]	R_S [%]	T_R [s]	R_R [%]	T_S [s]	R_S [%]
Algorithm 1												
76 176	4.01	100.00	4.92	100.00	21.04	100.00	22.12	100.00	103.32	100.00	104.31	100.00
235 224	12.85	100.00	14.42	100.00	67.98	100.00	69.72	100.00	261.55	100.00	263.24	100.00
710 976	37.05	100.00	40.83	100.00	219.61	100.00	223.49	100.00	–	–	–	–
2 154 720	112.97	100.00	123.92	100.00	599.72	100.00	610.71	100.00	–	–	–	–
Algorithm 2												
76 176	2.64	65.91	3.61	73.30	11.26	53.51	12.29	55.55	42.85	41.47	43.87	42.05
235 224	9.34	72.74	11.11	77.04	35.15	51.71	36.79	52.78	138.65	53.01	140.32	53.30
710 976	24.09	65.01	28.05	68.70	102.09	46.49	106.37	47.60	403.55	–	407.66	–
2 154 720	71.91	63.66	82.50	66.57	304.28	50.74	315.66	51.69	–	–	–	–
Algorithm 3												
76 176	1.85	46.21	2.78	56.44	3.32	15.78	4.24	19.15	5.13	4.96	6.07	5.81
235 224	5.30	41.26	6.95	48.19	9.06	13.33	10.71	15.36	17.19	6.57	18.87	7.17
710 976	18.04	48.69	22.33	54.68	28.69	13.06	32.54	14.56	54.53	–	58.77	–
2 154 720	49.22	43.57	59.59	48.08	89.54	14.93	100.62	16.48	156.97	–	167.89	–
Algorithm 4												
76 176	0.36	9.08	1.23	24.95	0.80	3.78	1.76	7.94	1.12	1.08	2.02	1.94
235 224	0.77	5.97	2.41	16.72	2.00	2.94	3.58	5.13	3.27	1.25	4.90	1.86
710 976	2.74	7.40	6.56	16.06	5.29	2.41	9.31	4.17	11.19	–	15.16	–
2 154 720	7.01	6.20	17.59	14.19	18.28	3.05	29.34	4.80	32.66	–	43.38	–

- For polynomial degree $d=3$, algorithms 2, 3, and 4 provide a relative memory usage of 88%, 26%, and 34%, respectively.
- For polynomial degree $d=5$, algorithms 2, 3, and 4 provide a relative memory usage of 57%, 8%, and 10%, respectively.

Algorithm 2 always requires a higher amount of memory for prismatic meshes than tetrahedral meshes due to the higher number of quadrature points and, therefore, more vectors need to be stored. On the other side, algorithms 3 and algorithm 4 give relative memory usage similar to the case of tetrahedral meshes since the number of vectors to store does not depend on the number of quadrature points.

5.6.3 Hexahedral meshes

Finally, simulations with successive finer hexahedral meshes, polynomial degrees $d=1,3,5$, and algorithms 1 to 4 are carried out. The residual time, solution time, and corresponding relative

Table 5.19: Memory usage and relative values obtained with prismatic meshes.

	$d = 1$		$d = 3$		$d = 5$	
DOF	M [Gb]	R_{M} [%]	M [Gb]	R_{M} [%]	M [Gb]	R_{M} [%]
Algorithm 1						
76 176	1.09	100.00	5.50	100.00	33.66	100.00
235 224	3.18	100.00	16.38	100.00	100.73	100.00
710 976	9.43	100.00	48.17	100.00	–	–
2 154 720	28.02	100.00	143.57	100.00	–	–
Algorithm 2						
76 176	1.01	92.29	4.75	86.49	19.00	56.46
235 224	3.09	97.15	14.21	86.76	57.48	57.07
710 976	9.04	95.83	42.07	87.34	171.05	–
2 154 720	26.85	95.82	126.09	87.82	–	–
Algorithm 3						
76 176	0.93	85.41	1.44	26.14	2.59	7.70
235 224	2.69	84.68	4.29	26.18	7.79	7.74
710 976	7.81	82.79	12.54	26.02	22.81	–
2 154 720	23.14	82.58	37.50	26.12	68.39	–
Algorithm 4						
76 176	1.01	92.93	1.91	34.70	3.48	10.33
235 224	2.94	92.51	5.57	33.99	10.37	10.29
710 976	8.52	90.28	16.50	34.25	30.70	–
2 154 720	25.34	90.44	49.49	34.47	91.90	–

values are reported in Table 5.20. Once again, with algorithms 1 and 2 and polynomial degree $d = 5$, the residual time and solution time are not provided for some meshes since the amount required for the simulation exceeds the available resources. Hence, the relative residual and solution time concerning algorithms 2, 3, and 4 can not be determined.

The residual time and solution time follow the case of tetrahedral meshes and prismatic meshes. In general, the relative residual time and solution time change substantially with the number of degrees of freedom, possibly due to cache effects or other implementation aspects that are difficult to control. Indeed, in some cases, the relative residual time and solution time increase with the number of degrees of freedom, whereas, it decreases in other cases. To sum up, the following computational gains in terms of execution time are obtained for the finest hexahedral mesh and depicted in Figure 5.13:

- For polynomial degree $d = 1$, algorithms 2, 3, and 4 provide a relative residual time of 60%,

40%, and 5%, respectively, and a relative solution time of 63%, 44%, and 12%, respectively.

- For polynomial degree $d = 3$, algorithms 2, 3, and 4 provide a relative residual time of 55%, 20%, and 2%, respectively, and a relative solution time of 56%, 21%, and 4%, respectively.
- For polynomial degree $d = 5$, algorithms 2, 3, and 4 provide a relative residual time of 62%, 6%, and 1%, respectively, and a relative solution time of 62%, 6%, and 2%, respectively.

For all the simulations, algorithm 4 gives a significant improvement for the second-, fourth-, and sixth-orders of convergence, with polynomial degrees $d = 1, 3, 5$, respectively, with relative solution time of 12% (8 times faster), 4% (25 times faster), and 2% (50 times faster), respectively. Moreover, in the general case and regardless of the algorithm, higher computational gains are achieved with hexahedral meshes than with tetrahedral meshes, but are more expressive with algorithms 3 and 4. Indeed, all the faces in prismatic meshes are quadrilaterals and, therefore, require twice the quadrature points of triangular faces to obtain the same convergence order of the numerical integration. Consequently, since more quadrature points are required in the quadrilateral faces of the hexahedral mesh elements to obtain the same convergence order, a more significant computational gain is expected.

As for prismatic meshes, the absolute residual time and solution time obtained with algorithm 4 for hexahedral meshes are comparable with the ones obtained with the same algorithm for tetrahedral meshes. These results further support the previous conclusions and emphasizes the benefits of an optimized implementation for the computational cost of the simulations.

The memory usage and corresponding relative values are reported in Table 5.21. Once again, with algorithms 1 and 2 and polynomial degree $d = 5$, the memory usage is not provided for some meshes since the amount required for the simulation exceeds the available resources. Hence, the relative memory usage concerning algorithms 2, 3, and 4 can not be determined.

To sum up, the following computational gains in terms of memory usage are obtained for the finest hexahedral mesh and depicted in Figure 5.13:

- For polynomial degree $d = 1$, algorithms 2, 3, and 4 provide a relative memory usage of 101%, 82%, and 89%, respectively.
- For polynomial degree $d = 3$, algorithms 2, 3, and 4 provide a relative memory usage of 111%, 27%, and 34%, respectively.
- For polynomial degree $d = 5$, algorithms 2, 3, and 4 provide a relative memory usage of 75%, 8%, and 10%, respectively.

Algorithm 2 with hexahedral meshes requires more memory than with tetrahedral and prismatic meshes due to the higher number of quadrature points. Indeed, for polynomial degree $d = 1$, algorithms 1 and 2 use a similar amount of memory, while the memory usage is exceeded for

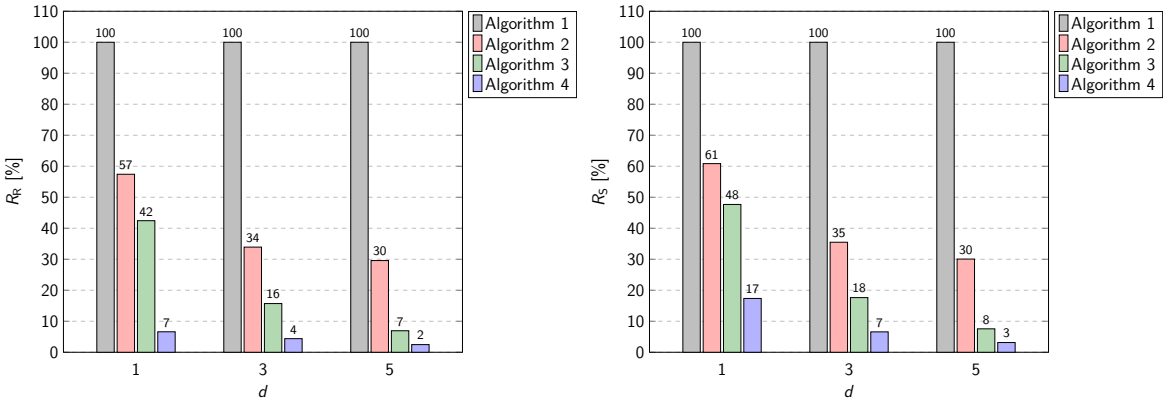
Table 5.20: Residual time, solution time, and relative values obtained with hexahedral meshes.

	$d=1$				$d=3$				$d=5$			
DOF	T_R [s]	R_R [%]	T_S [s]	R_S [%]	T_R [s]	R_R [%]	T_S [s]	R_S [%]	T_R [s]	R_R [%]	T_S [s]	R_S [%]
Algorithm 1												
80 736	5.58	100.00	6.59	100.00	24.65	100.00	25.70	100.00	126.97	100.00	128.01	100.00
254 016	16.18	100.00	17.97	100.00	93.99	100.00	95.88	100.00	345.66	100.00	347.48	100.00
726 624	46.90	100.00	50.98	100.00	277.23	100.00	281.63	100.00	–	–	–	–
2 254 200	146.59	100.00	158.32	100.00	–	–	–	–	–	–	–	–
Algorithm 2												
80 736	3.00	53.79	4.04	61.24	17.20	69.80	18.25	71.00	69.13	54.44	70.15	54.80
254 016	10.73	66.35	12.45	69.24	55.78	59.34	57.63	60.11	214.18	61.96	215.98	62.16
726 624	30.40	64.82	34.23	67.14	153.58	55.40	157.94	56.08	–	–	–	–
2 254 200	88.60	60.44	99.75	63.00	431.34	–	442.59	–	–	–	–	–
Algorithm 3												
80 736	2.11	37.88	3.03	45.93	4.42	17.95	5.44	21.16	6.51	5.13	7.45	5.82
254 016	6.67	41.25	8.43	46.88	11.85	12.61	13.63	14.22	20.50	5.93	22.28	6.41
726 624	18.80	40.08	22.85	44.82	54.79	19.76	58.74	20.86	61.05	–	65.14	–
2 254 200	58.71	40.05	69.24	43.73	112.34	–	124.04	–	203.45	–	215.75	–
Algorithm 4												
80 736	0.35	6.33	1.26	19.09	0.80	3.25	1.80	7.02	1.34	1.06	2.28	1.78
254 016	1.03	6.37	2.76	15.35	2.30	2.45	4.05	4.22	4.11	1.19	5.83	1.68
726 624	2.32	4.95	6.26	12.29	6.48	2.34	10.41	3.70	10.60	–	14.60	–
2 254 200	7.96	5.43	19.44	12.28	19.62	–	32.41	–	31.83	–	43.43	–

polynomial degree $d = 3$. Nevertheless, algorithms 3 and 4 with hexahedral meshes have comparable memory usage to tetrahedral and prismatic meshes as the size of the vectors to store does not depend on the type of mesh elements.

5.7 Conclusions

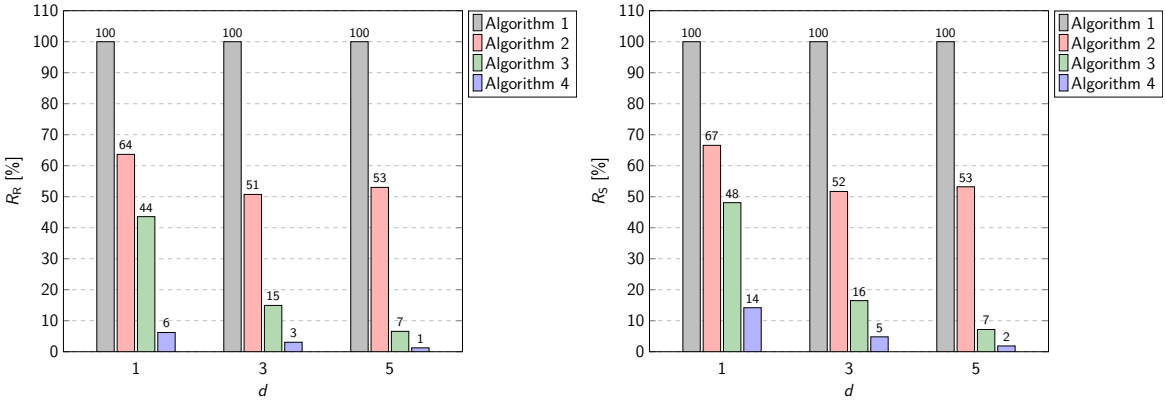
A very high-order accurate finite volume scheme is proposed to solve three-dimensional steady-state conjugate heat transfer problems in arbitrary curved domains with the continuity interface conditions. The conjugate problem model is partitioned with a Dirichlet-Neumann method that replaces the continuity interface conditions with Dirichlet and Neumann boundary conditions prescribed on the interface. The Robin boundary condition is used on both the boundaries and the interface as a general approach to prescribed the Dirichlet and Neumann ones assigning the appropriate coefficients. The partitioned problem model consists of subproblems in the physical subdomains with only boundary conditions and, therefore, are discretized separately with the proposed method. On the other side, the



(a) Relative residual time.

(b) Relative solution time.

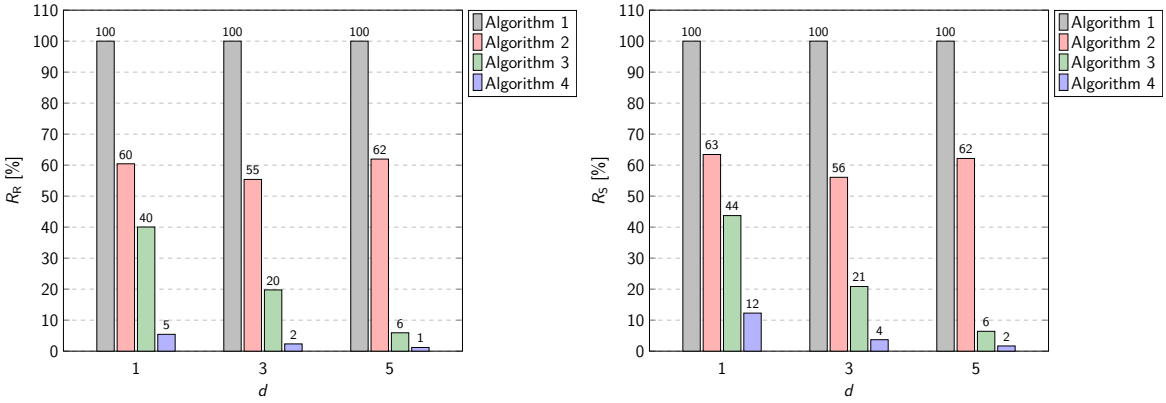
Figure 5.13: Relative residual time and solution time obtained with tetrahedral meshes.



(a) Relative residual time.

(b) Relative solution time.

Figure 5.14: Relative residual time and solution time obtained with prismatic meshes.



(a) Relative residual time.

(b) Relative solution time.

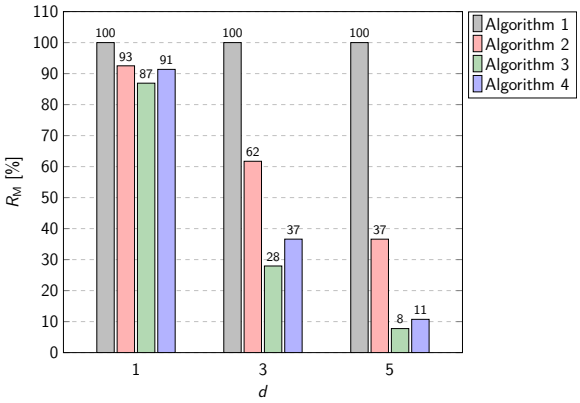
Figure 5.15: Relative residual time and solution time obtained with hexahedral meshes.

Table 5.21: Memory usage and relative values obtained with hexahedral meshes.

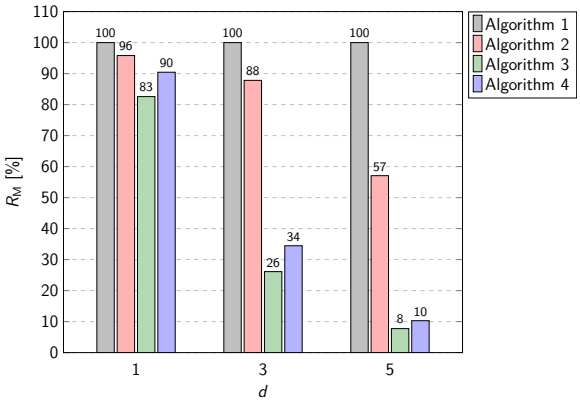
	$d=1$		$d=3$		$d=5$	
DOF	M [Gb]	R_{M} [%]	M [Gb]	R_{M} [%]	M [Gb]	R_{M} [%]
Algorithm 1						
80 736	1.26	100.00	6.43	100.00	39.50	100.00
254 016	3.91	100.00	19.83	100.00	121.68	100.00
726 624	11.12	100.00	55.60	100.00	–	–
2 254 200	33.96	100.00	–	–	–	–
Algorithm 2						
80 736	1.27	101.09	7.11	110.53	29.64	75.04
254 016	3.99	102.06	22.02	111.09	91.33	75.06
726 624	11.13	100.11	61.46	110.54	–	–
2 254 200	33.99	100.09	189.11	–	–	–
Algorithm 3						
80 736	1.06	84.07	1.72	26.81	3.18	8.05
254 016	3.27	83.77	5.30	26.72	9.61	7.90
726 624	9.11	81.90	14.74	26.52	27.15	–
2 254 200	27.82	81.91	45.26	–	83.64	–
Algorithm 4						
80 736	1.19	94.26	2.22	34.50	4.00	10.12
254 016	3.50	89.59	6.64	33.51	12.44	10.22
726 624	9.82	88.27	18.73	33.68	34.95	–
2 254 200	30.06	88.52	57.78	–	107.43	–

thermal coupling between subproblems consists in providing the appropriate values to the boundary conditions on the interface such that the heat transfer is recovered.

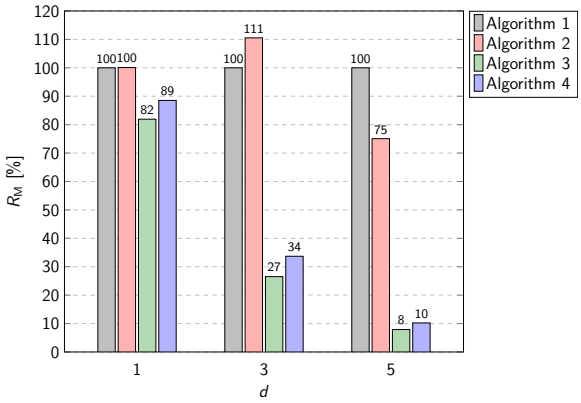
The very high-order finite volume scheme discretizes each subproblem and consists of polynomial reconstructions that provide local approximations of the temperature functions. Specific polynomial reconstructions are computed to determine the numerical heat fluxes on the faces and also to provide the proper relations of the boundary variables for the thermal coupling. Polynomial reconstructions associated with the boundary faces are computed applying specific linear constraints derived from the Robin boundary condition. In that regard, the ROD method avoids the accuracy deterioration and the second-order of convergence limitation observed with the conventional methods on arbitrary curved boundaries and interfaces. On the other side, solely polyhedral meshes are used in the ROD method, therefore, overcoming the difficulties associated with the methods relying on curved mesh elements, such as the isoparametric elements method. A simple implementation of the proposed method



(a) Tetrahedral meshes.



(b) Prismatic meshes.



(c) Hexahedral meshes.

Figure 5.16: Relative memory usage.

provides an implicit system of linear equations through a global residual operator, which is solved with free-matrix iterative algorithms, such as the GMRES method. Four algorithms are proposed to gradually optimize the implementation, reducing the computational cost and improving the efficiency of the proposed method. The proposed algorithms are derived from the algebraic rearrangement of the proposed method without introducing more approximations at the cost of an additional, but small, pre-processing stage. In particular, the latest algorithm provides an explicit system of linear equations through the conventional coefficients matrix and associated right-hand side, which can be solved with any suitable method.

A comprehensive numerical benchmark is provided for the verification of the proposed method in terms of accuracy and convergence orders based on the method of manufactured solutions. Several test cases are addressed consisting of different domains, boundary conditions, thermal conductivity, velocity, for which specific analytic solutions are derived. The proposed method achieves the optimal convergence orders effectively and is capable of handling general polyhedral meshes generated for arbitrary curved domains. Moreover, the partitioning configuration impacts the computational efficiency as it strongly determines the number of iterations in the GMRES method, particularly for high thermal conductivity ratios and polynomial degrees. More precisely, prescribing the Dirichlet boundary

condition in the subproblem with the lower thermal conductivity (and the Neumann boundary condition in the other) requires fewer iterations and, therefore, is computationally more efficient. Nevertheless, comparable accuracy and convergence orders are obtained with both partitioning configurations.

A comprehensive computational benchmark is provided for the assessment of the computational gain in terms of execution time and memory usage provided with the proposed algorithms. The computational gain gradually increases from algorithm 1 to 4, both in terms of execution time and memory usage, and is particularly relevant for higher polynomial degrees. The explicit formulation (algorithm 4) provides substantial computational gains up to 1% of the execution time and up to 11% of the memory usage relative to algorithm 1. The results prove the importance of the implementation to drastically reduce the computational cost and provide the same accuracy with less demanding simulations in terms of computational resources. Furthermore, from a sustainability perspective, very high-order accurate methods supplied with efficient implementations are promising means of reducing the energy expenditure.

The proposed developments are promising steps towards efficient simulations of conjugate heat transfer problems and multiphysics problems in complex engineering applications, where verification and validation in real scenarios with real parameters will be required. The proposed method is very versatile and can be extended to solve other equations, such as the Navier-Stokes, whereas the implementation improvements will be applied similarly.

Appendices

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, weighted cost functional (5.15) is defined and given in matrix form as

$$F(\boldsymbol{\eta}) = \left\| \boldsymbol{W}\boldsymbol{A}\boldsymbol{\eta} - \boldsymbol{W}\boldsymbol{b} \right\|_2^2, \tag{5.147}$$

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, given as in Equation (5.16), or to fulfill the Robin boundary condition, given as in Equation (5.17). In both cases, constraint functional $G(\boldsymbol{\eta})$ is

given in matrix form as

$$G(\boldsymbol{\eta}) = \boldsymbol{c} \cdot \boldsymbol{\eta} - g, \tag{5.148}$$

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 fulfill, 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 $F(\boldsymbol{\eta})$ in the least-squares sense, that is, $\widetilde{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} F(\boldsymbol{\eta})$. Following the normal equations method [16], vector $\widetilde{\boldsymbol{\eta}}$ can be computed given as

$$\widetilde{\boldsymbol{\eta}} = (\boldsymbol{WA})^{\dagger}\boldsymbol{Wb} = \left((\boldsymbol{WA})^{\mathrm{T}}\boldsymbol{WA}\right)^{-1}(\boldsymbol{WA})^{\mathrm{T}}\boldsymbol{Wb} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} \tag{5.149}$$

where $(\boldsymbol{WA})^{\dagger}$ corresponds to the Moore-Penrose pseudoinverse of matrix $\boldsymbol{WA}$ (notice that $\boldsymbol{W}^{\mathrm{T}} = \boldsymbol{W}$ since it is a diagonal matrix). Considering matrix $\boldsymbol{M} \in \mathbb{R}^{s \times n}$ given as $\boldsymbol{M} = (\boldsymbol{WA})^{\dagger}\boldsymbol{W}$, then polynomial coefficients vector is given as $\widetilde{\boldsymbol{\eta}} = \boldsymbol{Mb}$. 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 numbers 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_{Δ}^{S}, associated polynomial coefficients vector $\widetilde{\boldsymbol{\eta}}_{ij}$ can be computed given 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 (5.149).

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 $F(\boldsymbol{\eta})$ in the least-squares sense and exactly fulfills 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 [17], consider functional $L(\boldsymbol{\eta}, \lambda)$ given as

$$L(\boldsymbol{\eta}, \lambda) = F(\boldsymbol{\eta}) + \lambda G(\boldsymbol{\eta}) \tag{5.150}$$

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 $\widehat{\boldsymbol{\eta}}$ can be computed from the the solution of system of linear equations given as $\nabla_{\boldsymbol{\eta},\lambda} L(\boldsymbol{\eta},\lambda) = \boldsymbol{0}$ 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

$$\nabla_{\boldsymbol{\eta}} F(\boldsymbol{\eta}) = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A}\boldsymbol{\eta} - 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b}, \tag{5.151}$$

$$\nabla_{\boldsymbol{\eta}} G(\boldsymbol{\eta}) = \boldsymbol{c}, \tag{5.152}$$

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

$$\nabla_{\boldsymbol{\eta}} L(\boldsymbol{\eta},\lambda) = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A}\boldsymbol{\eta} - 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} + \lambda\boldsymbol{c}. \tag{5.153}$$

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 (5.148). 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

$$\begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A} & \boldsymbol{c} \\ \boldsymbol{c}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} \\ g \end{bmatrix}, \tag{5.154}$$

and, therefore, polynomial coefficients vector $\widehat{\boldsymbol{\eta}}$ and associated Lagrange multiplier $\widehat{\lambda}$ are given as

$$\begin{bmatrix} \widehat{\boldsymbol{\eta}} \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{A} & \boldsymbol{c} \\ \boldsymbol{c}^{\mathrm{T}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^2\boldsymbol{b} \\ g \end{bmatrix}. \tag{5.155}$$

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 (5.154), an LDLT factorization (a closely related variant of the conventional Cholesky factorization, also called LLT factorization) is used since the associated coefficients matrix is symmetric by construction. The Cholesky factorizations perform roughly twice as efficient as the LU factorization if optimally implemented and requires half of the memory with sparse matrix storage [22]. 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 one) and, therefore, matrices $\boldsymbol{D}$ and $\boldsymbol{L}$ can be stored 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 numbers of associated coefficients 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 (5.155). 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 (5.155). Notice that, in that case, $g = g_{iF}$ is a point-value and, therefore, is not gathered in vector $\boldsymbol{b}_{iF}$.

References

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

[2] 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.

[3] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.-O. Persson, B. Leer, M. Visbal, High-order CFD methods: current status and perspective, Int. J. Numer. Meth. Fluids 72 (2013) 811–845.

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

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

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

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

[8] C. Ollivier-Gooch, A. Nejat, C. Michalak, On obtaining high-order finite-volume solutions to the Euler equations on unstructured meshes, AIAA Paper 2007-4464 (2007).

[9] C. Ollivier-Gooch, A. Nejat, C. Michalak, Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations, AIAA Journal 47 (9) (2009) 2105–2120.

[10] C. Michalak, C. Ollivier-Gooch, Unstructured high-order accurate finite volume solutions of the Navier-Stokes equations, AIAA Paper 2009-954 (2009).

[11] A. Boularas, S. Clain, F. Baudoin, A sixth-order finite volume method for diffusion problem with curved boundaries, Appl. Math. Model. 42 (2017) 401–422.

[12] J. Fernández-Fidalgo, S. Clain, L. Ramírez, I. Colominas, X. Nogueira, Very high-order method on immersed curved domains for finite difference schemes with regular Cartesian grids, Comput. Meth. Appl. Mech. Engrg. 360 (2020) 112782.

[13] 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.

[14] R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, R. Loubère, Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries, Int. J. Numer. Meth. Engrg. 117 (2) (2019) 188–220.

[15] R. Costa, S. Clain, R. Loubère, G.J. Machado, Very high-order accurate polygonal mesh finite volume scheme for conjugate heat transfer problems with curved interfaces and imperfect contacts, Comput. Meth. Appl. Mech. Engrg. 357 (2019) 112560.

[16] D.E. Wells, E.J. Krakiwsky, The method of least-squares, lecture notes 18, Department of Surveying Engineering, University of New Brunswick, Fredericton (1971).

[17] D.P. Bertsekas, Constrained optimization and Lagrange multiplier methods, 1st Edition, Academic Press (1982).

[18] P.J. Roache, Code verification by the method of manufactured solutions, J. Fluids Engrg. 124 (1) (2002) 4–10.

[19] C.J. Roy, Grid convergence error analysis for mixed-order numerical schemes, AIAA Journal 41 (4) (2003) 595–604.

[20] C.J. Roy, C.C. Nelson, T.M. Smith, C.C. Ober, Verification of Euler/Navier-Stokes codes using the method of manufactured solutions, Int. J. Numer. Meth. Fluids 44 (2004) 599–620.

[21] A. Veeraragavan, J.Beri, R.J. Gollan, Use of the method of manufactured solutions for the verification of conjugate heat transfer solvers, J. Comput. Phys. 307 (2016) 308–320.

[22] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B,P. Flannery, Numerical recipes in C: The art of scientific computing, 2nd Edition, Cambridge University England EPress (1992).

CHAPTER 6 Conclusions and future work

The present thesis is completed with the general conclusions and some perspectives of future works, bearing in mind the achieved developments and the results presented in the previous chapters. Notice that, the structure of the thesis is adapted from publications, each addressing one or more questions within the proposed objectives and, as usual, also include the respective conclusions and proposals for future works at the end.

6.1 Conclusions

The different studies developed along the thesis are summed up in this section with some additional conclusions, and revisited in line with the proposed objectives for the present work, leaving aside the technical details.

6.1.1 General boundary conditions on curved boundaries

A finite volume method with a very high-order of convergence was proposed to solve the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries. An efficient numerical technique was developed to overcome the difficulties motivated by the geometrical mismatch between polygonal meshes and curved boundaries, which often leads to accuracy and convergence order deterioration. The technique is based on constrained polynomial reconstructions fulfilling the prescribed boundary conditions on the curved boundary, for which linear constraints in the associated least-squares fittings are employed. Then, the numerical approximations of the convective and conductive physical fluxes, derived from the generic finite volume formulation, are determined solely on the cells polygonal boundaries. In that way, the proposed numerical technique overcomes the drawbacks associated with the classical techniques relying on curved mesh elements, such as the isoparametric elements method. For instance, the elaborated and computationally expensive algorithms to generate curved meshes, the complex non-linear transformations, and the quadrature rules on curved elements are avoided with the proposed approach. These requirements are particularly challenging in realistic three-dimensional geometries, whereas it becomes feasible and practical in the

case of the proposed method.

Exhaustive verification of these developments was carried out with numerical benchmarks comprising several test cases with non-trivial curved boundaries, prescribed with the Dirichlet, Neumann, and Robin boundary conditions. The results obtained prove that the proposed method effectively achieves the very high-order of convergence, whereas just the second-order of convergence is achieved when curved boundaries are handled without the appropriate treatment. Moreover, the very high-order of convergence approach provides significantly more accurate approximate solutions with the same mesh, when compared with the second-order of convergence counterparts. From another perspective, the proposed method also improves the performance of the calculation in terms of computational resources. That is, the very high-order of convergence provides the same accuracy with substantially coarser meshes, when compared with the second-order of convergence, therefore requiring less computer memory. The difference is more pronounced with the increasing demand for accuracy from the industrial applications, which can be infeasible with the second-order of convergence due to the limited available computational resources.

6.1.2 Conjugate heat transfer problems with general interface conditions

The proposed finite volume method with a very high-order of convergence was extended to conjugate heat transfer problems with general interface conditions on arbitrary curved interfaces. The method consists in partitioning the conjugate problem with the Dirichlet-Neumann or Neumann-Neumann methods, such that the interface conditions are replaced with specific boundary conditions on the interface. Then, each subproblem used only the conventional boundary conditions and, therefore, is discretized separately applying the previous numerical technique to preserve the convergence order with the use of polygonal meshes. The appropriate thermal coupling between subproblems consists in determining approximations of the boundary condition functions prescribed on the interface, with polynomial reconstructions computed based on the solution of the adjacent subproblem.

The verification of the proposed method was carefully carried out with numerical benchmarks comprising several test cases, with non-trivial curved interfaces prescribed with the continuity and imperfect interface conditions. The results obtained support the capability of the method to effectively achieve the very high-order of convergence in all the addressed test cases. Moreover, the temperature jumps arising on the interface, due to the interfacial thermal resistance of imperfect thermal contacts, are handled without any difficulties, preserving the convergence order of the method.

6.1.3 Thermal boundary layers in high-Péclet number problems

Several test cases concerning high-Péclet number problems and thermal boundary layers were addressed in the numerical benchmarks provided. The results obtained support the capability of the

proposed method to effectively achieve the very high-order of convergence, without any additional requirements, presenting an essentially non-oscillatory behaviour. Nevertheless, the verification of the method in an industrial representative application of relevant and complex heat transfer was the ultimate challenge to illustrate the practical benefits. In that regard, the method was applied to address the thermoplastic sheet extrusion cooling stage, an essential application in the polymer industry. A strong thermal boundary layer arises due to the temperature shock at the location where the initial contact between the melted thermoplastic sheet, leaving the extruder, and the chilled roll occurs. Moreover, due to the typically high extrusion speeds and the low thermal conduction of thermoplastics, the heat transfer in the problem is predominantly convective, having a high-Pléclet number and giving rise to sharp boundary layers.

The use of stretched meshes was proposed to reduce the total number of cells in the mesh since the temperature variations occur mainly along the perpendicular to the extrusion direction, handling this problem more efficiently. The results obtained prove that the method is capable of achieving the very high-order of convergence regardless of the employed mesh. However, as anticipated, accuracy deterioration is observed with increasing mesh aspect ratio. Therefore, there is a trade-off between accuracy and efficiency, which mainly depends on the application requirements. In the case of the thermoplastic sheet extrusion cooling stage, the very high-order of convergence method provides significantly more accurate solutions, when compared with the second-order of convergence method. On the other side, the latter is also significantly more time-consuming to achieve the same accuracy level. Indeed, the results prove that higher aspect ratios allow further reducing the computational cost of the simulations, considering the accuracy requirements of this specific application.

6.1.4 Three-dimensional domains and algorithm optimization

The extension of the proposed method to three-dimensional geometries raises an additional challenge with respect to the boundary description and the computational effort, when compared with the two-dimensional case. Indeed, there is a significantly higher computational cost as a direct consequence of the exponential dependence with the number of dimensions. With this motivation, several optimized algorithms were derived to improve the efficiency of the proposed method, particularly in the three-dimensional case, which often requires millions of cells and, thus, very large matrices. The proposed algorithms consist in successive algebraic simplifications, which avoid repetitive calculations in each iteration of the iterative method employed to solve the implicit system of linear equations. Then, these calculations are transferred to a pre-processing stage, improving the overall performance of the proposed method.

The verification of the proposed algorithms was carried out with computational benchmarks comprising several test cases in three-dimensional geometries, both with non-trivial curved boundaries and interfaces. First, the results obtained support the simplicity of the proposed method to handle the three-dimensional case and also the capability to provide a very high-order of convergence. Moreover,

the optimized algorithms drastically reduce the computational cost of the proposed method, both in terms of execution time and memory usage, without accuracy deterioration.

6.2 Future work

The objectives outlined for the present work were properly addressed, which provided valuable answers in the quest of high-performance simulations for polymer processing applications. The undertaken research proved that advanced computational methods are a very promising approach to provide more efficient simulations and more accurate solutions of complex problems, when compared with the classical counterparts. Indeed, many software companies are turning their attention to having more modern and sophisticated computational methods in their simulation products, which ultimately will benefit the industry with more capable and powerful tools.

In practical terms, the present work tackles some of the fundamental, but indispensable, questions for the development of advanced computational methods towards high-performance simulations. Nevertheless, there are plenty of other questions that remain open, and future works should be undertaken, aiming at the application of these methods to solve real context problems in the polymer processing industry. In that regard, the knowledge acquired in the present work provides solid foundations and useful guidelines that will be essential for further research on the topic.

Bearing in mind the previous thoughts, future works on the research topic should mainly concern the extension of the proposed methods to solve the classical models in fluid flow problems. For instance, the Navier-Stokes equations, widely used to model flow problems in a variety of situations, can be taken as the next challenge. Indeed, the model already raises many questions concerning the non-linearity and, for the case of incompressible fluid flows, the divergence-free constraint that should be considered. Associated with the Navier-Stokes equations, several situations become a challenge by themselves. For instance, boundary conditions taking into account the effect of the boundary curvature, interface conditions imposing the thermal exchanges between phases, and extreme boundary layers arising from high-Péclet number flows. Providing that these methods are still capable of improving performance in such a scenario, when compared with the classical ones, more complex and realistic models for the polymer processing context can be considered. For instance, the Cauchy equations with the appropriate constitutive equations for the polymeric fluid can be addressed, where the rheology will have an important role. The increased complexity of these models will undoubtedly bring additional challenges in preserving the accuracy of the approximate solutions and the computational efficiency of the simulation. The methods will eventually become more specialized to the complex models they are intended to solve, which requires a particularly significant effort on preserving versatility and simplicity.

The present work represents important steps towards high-performance polymer processing simulations, but still requires a plethora of developments, for which further research on this topic is necessary.