Imperial College London

Faculty of Engineering

Department of Aeronautics

Mesh Adaptation for High-Order Flow Simulations

A thesis submitted for the degree of

Doctor of Philosophy

Julian Marcon

Supervisors:

Prof. Joaquim Peiró

Prof. Spencer J. Sherwin

2020

Abstract

Mesh adaptation has only been considered for high-order flow simulations in recent years and many techniques are still to be made more robust and efficient with curvilinear meshes required by these high-order methods. This thesis covers the developments made to improve the mesh generation and adaptation capabilities of the open-source spectral/*hp* element framework *Nektar++* and its dedicated mesh utility *NekMesh*.

This thesis first covers the generation of quality initial meshes typically required before an iterative adaptation procedure can be used. For optimal performance of the spectral/*hp* element method, quadrilateral and hexahedral meshes are preferred and two methods are presented to achieve this, either entirely or partially. The first method, inspired from cross field methods, solves a Laplace problem to obtain a guiding field from which a valid two-dimensional quadrilateral block decomposition can be automatically obtained. In turn, naturally curved meshes are generated. The second method takes advantage of the medial axis to generate structured partitions in the boundary layer region of three-dimensional domains. The method proves to be robust in generating hybrid high-order meshes with boundary layer aligned prismatic elements near boundaries and tetrahedral elements elsewhere.

The thesis goes on to explore the adaptation of high-order meshes for the simulation of flows using a spectral/*hp* element formulation. First a new approach to moving meshes, referred to here as r-adaptation, based on a variational framework, is described. This new r-adaptation module is then enhanced by p-adaptation for the simulation of compressible inviscid flows with shocks. Where the flow is smooth, p-adaptation is used to benefit from the spectral convergence of the spectral/*hp* element methods. Where the flow is discontinuous, e.g. at shock waves, r-adaptation clusters nodes together to better capture these field discontinuities. The benefits of this dual, rp-adaptation approach are demonstrated through two-dimensional benchmark test cases.

Declaration of originality

I present this thesis for consideration for the degree of Doctor of Philosophy at Imperial College London exclusively. I certify that it contains the results of my own work and research during my doctoral studies, unless otherwise stated. In such cases, I give proper acknowledgement for methodologies and data that are not my own.

Copyright declaration

Acknowledgements

First and foremost, I am immensely grateful to my supervisors, Prof. Joaquim Peiró and Prof. Spencer J. Sherwin for the trust they gave me at recruitment. This has been the opportunity of a lifetime and one that defines a researcher. They have guided me and encouraged me throughout this whole process. They have also pushed me and challenged me to become the best researcher I could be. I believe I have become a full member of the high-order mesh generation and adaptation community and I owe it to them.

I would also like to thank the broader *Nektar++/NekMesh* research group. Of everyone, I would like to mostly thank Dr. David Moxey for his help and guidance throughout the course of this research. His insight was invaluable and I wish him plenty of success in his academic career. I have also received important support at various stages from the following people whom I would like to acknowledge: Dr. Michael Turner, Dr. Chris Cantwell and Dr. Giacomo Castiglioni.

I am delighted to have also had so many chances to collaborate with external partners. I was first able to work with industrial partners of the International TechneGroup and would like to thank Mark Gammon, Henry Bucklow, Claire Pollard and, more recently, Dr. Nico Bergemann for their support. I have also had the chance to visit ONERA, the French Aerospace Lab, in Paris, France, and would like to thank Dr. Fabio Naddei, Dr. Vincent Couaillier and Dr. Marta de la Llave Plata for hosting me and providing this chance to explore new ideas. Finally I would like to thank Prof. David A. Kopriva for hosting my secondment in San Diego, CA, USA. We have made tremendous progress together and I have gained invaluable experience from working with him, in terms of both technical and writing skills.

This research would not have been possible without financial support. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675008. I have also received funding from PRISM (Platform for Research In Simulation Methods) under EPSRC grant EP/L000407/1. I would also like to thank the committee of the 28th International Meshing Roundtable (IMR28) for their financial

support towards the travel costs of attending the conference.

My last few lines are dedicated to people closer to me personally. First I want to thank my parents for always supporting me in all my endeavours, both academic and and extra-curricular. I would also like to thank my office friends for having made my stay at Imperial College a socially delightful experience too. My final thanks go to Laura for her support during these past few months of intensive writing.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The accurate and high-fidelity simulation of high-speed compressible flows is, at present, a problem of significant interest to the aeronautics community, particularly in relation to aviation in which such conditions are routinely encountered. The complex and interdependent fluid phenomena found in this regime pose a difficult challenge for numerical modelling, with a stark contrast between regions of smooth flow, boundary layers near solid walls where large velocity gradients are present, and shock waves or shear layers where the fluid properties change sharply in a discontinuous manner.

The use of high-order spectral/*hp* element methods in the simulation of compressible fluid dynamics is now becoming increasingly common for high-fidelity large-eddy simulations and direct numerical simulations of realistic aeronautical configurations [9, 29, 86, 71, 87]. As in traditional low-order methods, the domain of interest is partitioned into finite elements; however these elements are also equipped with high-order polynomial expansions, as opposed to traditional linear shape functions.

1.1 Objectives

This high-order discretisation yields several advantages in terms of computational performance, as well as enhanced numerical resolution as p is increased. However, in the presence of shocks and discontinuities, the latter advantage will not be realised and can lead to significant issues in terms of stability and accuracy in the resolution of shocks. A common approach used in the resolution of discontinuous features is to refine these regions in an adaptive manner, so that the mesh resolution around the features is increased. In broad terms, the error of a computed solution which is sufficiently smooth can be roughly expressed as $\epsilon \approx Ch^p$, where C is a constant related to the measure of the solution regularity, h is the mesh size, and p is the polynomial order.

Mesh adaptation is concerned with achieving increased resolution by either locally reducing the mesh size h or locally increasing the polynomial order p. Due to its higher convergence rates, p-adaptation is typically preferred over h-adaptation for smooth flow regions [15, 40, 24], whereas the opposite is true where flow discontinuities exist [1, 59]. The reason for the latter — h-adaptation being preferred for flow discontinuities — is that the representation of shocks by high-order discretisations leads to numerical oscillations that must be smoothed out by the addition of high-order dissipation terms. This effectively means that the high-order DOF[1] are wasted in the vicinity of shocks. This thesis aims to achieve a combination of and balance between h- and p-type adaptation, enabled by spectral/hp element methods.

To obtain h-type adaptation, this work describes a moving mesh approach, referred to here as r-adaptation, based on a variational framework, where DOF are relocated within the domain. By using r- instead of h-adaptation, mesh connectivities are preserved, opening the way to transient simulations where the system of equations would not need to be re-built at each adaptation step. The DOF count additionally remains constant for better control of the computational cost of the simulation.

Adaptation methods, however, rely on a quality initial mesh as a starting point but the generation of valid high-order meshes remains one of the bottlenecks of the high-order simulation of high-fidelity flows [85, 86, 71, 78, 87]. The main challenge is to systematically and robustly generate valid high-order curvilinear boundary-conforming meshes that incorporate stretched elements in the near-wall boundary layer regions. If the complexity of the computational domain lends itself to structured multi-block decomposition [3], then the mapping between the blocks and the unit cube provided by this approach facilitates high-order and boundary layer meshing, but domain decomposition for general domains remains a very difficult and open problem. Where a block structure is not possible or hard to obtain, unstructured meshes are usually preferred because they are easy to generate. Their extension to high-order is typically based on *a posteriori* approaches that deform a coarse linear mesh to accommodate the curvature at the boundary; see, for instance, a brief review of these methods in reference [80]. Robust mesh generators are available for generating the linear mesh, but *a posteriori* high-order mesh generators of curvilinear meshes tend to have difficulties in ensuring the validity of the mesh when highly stretched elements, typical of boundary layer meshes, are present.

In two dimensions, although still difficult, important advances have been made

[1] DOF: degrees of freedom.

towards fully automatic quadrilateral block decomposition of arbitrary domains [72, 8]. One promising approach involves cross fields, which provide a guiding field for the decomposition of arbitrary domains into quadrilateral meshes. Most previous work has focussed on generating large numbers of small straight sided elements. The extension to high-order is usually done by curving the elements *a posteriori* [30]. This thesis presents an approach that uses this guiding field to generate coarse quadrilateral meshes that are naturally curved and valid. These can be further divided using an isoparametric approach that guarantees the validity of the final high-order mesh.

The extension of this method to three dimensions is, however, impractical due to the lack of automatic methods to generate frame — three-dimensional cross — fields that yield valid hexahedral block decompositions in a robust manner. For this reason, a semi-structure approach is presented in this thesis that combines two complementary mesh generation procedures. Because valid high-order mesh generation is difficult near curved boundaries, the domain is split into a near-field and a far-field partitions. The near-field partition is robustly created using a medial object approach, yielding a coarse linear boundary layer mesh. This coarse mesh can be *a posteriori* curved with the thickness of the layer guaranteeing enough space for the valid curving of boundary faces. The resulting high-order boundary layer mesh can be further refined to obtain valid, highly stretched elements capable of resolving boundary layer flows. The rest of the domain is subsequently discretised using traditional unstructured mesh generation approaches.

To achieve these objectives, the *Nektar++* platform [17, 50] was used. *Nektar++* is an open source, spectral/*hp* element environment used for the high-fidelity simulation of compressible and incompressible flows. It is accompanied by *NekMesh* [79, 50], a suite of tools dedicated to the generation and manipulation of high-order meshes. The work presented in this thesis relates to developments of the author in this combined codebase, unless otherwise stated.

1.2 Two-dimensional quadrilateral meshes

A quadrilateral subdivision of a two dimensional domain into a minimal number of subdomains can serve as the starting point from which to generate meshes for finite element [83], finite volume, (block structured) finite difference [38] and spectral/*hp* element methods [35, 36]. To generate finite element and finite volume meshes, one would subdivide each subdomain into quadrilateral cells sufficiently small to resolve the geometry and expected solution features. The resulting mesh would have the desired property of only a small number of irregular nodes where the valence, i.e. the number of elements sharing a node, is not equal to four. The subdivision of the subdomains could also be used as basis for high-order block-structured finite difference methods if the subdivision within each block is structured. Finally the subdomains could be used *as is* with high-order geometry information intact, or subdivided in a structured or unstructured manner, to be used as elements for a quadrilateral-based spectral/*hp* element method.

Unstructured quadrilateral mesh generators, on the other hand, tend to generate meshes with large numbers of irregular nodes [75]. Extraneous irregular nodes are generated even for simple geometries, where mesh topologies can be generated by hand with a minimum number of them. As an example, Fig. 1.1 shows two meshes generated with a subdivision algorithm [41, 67], which generates more subdivisions than necessary. Instead, a simpler block decomposition is sought with fewer irregular nodes and fewer blocks, offering more flexibility for further subdivision while retaining better mesh quality.

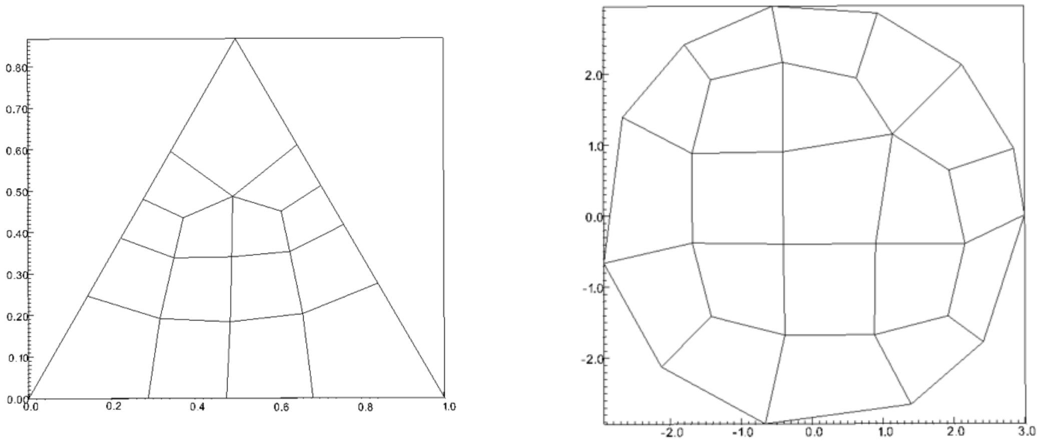


Figure 1.1: Quadrilateral meshes on simple geometries using a subdivision algorithm.

One robust approach for automatically generating quadrilateral block decompositions of arbitrary domains has come out of the computer graphics community. It uses cross fields [8, 83] in a field-guided procedure. Cross fields resemble cross hatchings used in drawings. Crosses, which are composed of two direction vectors and their negatives at a point, are invariant to rotations of $\pi/2$. Cross field methods generate meshes with fewer irregular nodes. Fig. 1.2 shows the same two geometries as Fig. 1.1 decomposed with a cross field method. The method produces the same topology as an experienced user might generate by hand, and has a minimum number of irregular nodes.

Details of cross field procedures vary, but the basic idea is that the crosses are

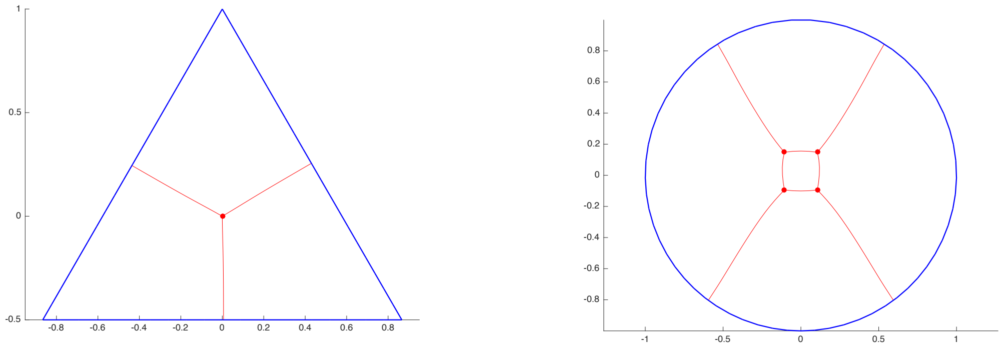


Figure 1.2: Quadrilateral decompositions on simple geometries using a cross field approach: courtesy of Ms. Bing Yuan.

oriented, usually at the nodes of an existing triangular mesh, according to an energy minimisation or smoothing procedure. Triangles in which the crosses cannot smoothly vary contain singularities, which serve as the irregular nodes of the quadrilateral decomposition to be computed. The internal edges of the quadrilaterals are found by integrating *separatrices* (represented as *streamlines* of the cross field) [37], starting from the irregular nodes until they reach another irregular node or a physical boundary. The quadrilateral subdivision is usually further subdivided into smaller quadrilateral elements.

Traditional techniques using crosses do not precisely locate irregular nodes within the domain. Irregular nodes can be located only to within the size of the element in which they occur. Spurious singularities can be generated, which may have to be coalesced in an *ad hoc* fashion. For instance, if the underlying mesh is too coarse, it is possible that two or more singularities fall within one element. The valence of the irregular nodes computed from a coalesced singularity could lead to the incorrect number of separatrices and hence the incorrect valence of the associated interior quadrilateral mesh node.

The traditional tracing of the separatrices also often leads to multiple curves that need to be coalesced. Associated with this is the well-known limit cycle problem where separatrices, instead of meeting as they should, spiral indefinitely [84].

Finally, BC[2] are difficult to apply at corners where a discontinuity in the cross field will occur when a standard continuous Galerkin (finite element) approximation is used for smoothing. Rather than putting a singularity there, the corner may be effectively smoothed by averaging the cross field at such points [83], at the expense of adding more interior singular points.

[2] BC: boundary conditions.

Issues found in the use of traditional techniques are addressed by using a high-resolution approach to compute a guiding field, accurately locating the interior irregular nodes, and accurately integrating the streamlines used to subdivide the domain. A high-order continuous or discontinuous Galerkin spectral/*hp* element method is used on a triangular mesh, the approximation depending on the regularity of the BC needed to solve Laplace problems. Unlike previous work, all operations are performed on the original highly-resolved guiding field rather than a cross field, so that accuracy is not lost. An accurate locator is used for the irregular points, that exploits the high-resolution solution. An accurate method is also used for computing their valence. The streamline integration then uses the high-order information available to obtain precise streamlines, which reduces the need to coalesce extraneous lines. A naturally curved, valid quadrilateral mesh is finally obtained that can be further subdivided using an isoparametric approach.

1.3 Three-dimensional hybrid meshes

Because the automatic block decomposition of three-dimensional domains is difficult, other techniques have been developed to analyse the shape of these geometrical domains. The difficulty especially resides in the generation of high aspect ratio elements in the boundary layer regions, where curvature yields invalid high-order elements. A generic approach consists of optimising the entire high-order mesh with respect to a measure of its deformation [78]. This method, while robust, can be slow and expensive. Likewise, an elastic analogy is often used [52] to propagate the boundary curvature to the interior of the domain. Both procedures, however, struggle at keeping highly anisotropic boundary layer elements valid. Because of this difficulty, some tools have become available specifically for high-order boundary layer mesh generation [43, 66, 62]. All of these have, however, looked at curving boundary layer elements *a posteriori*. Ch. 4 aims at generating a highly anisotropic boundary layer mesh only after obtaining a coarse valid mesh, i.e. *a priori* curving.

One promising technique, introduced by Blum [7], is the medial axis, which is defined as the set of all points in the domain that have more than one closest point on the boundary of the domain. The medial axis, along with other proximity information, provides a complete description of the geometry that can be used to generate partitions. This thesis proposes to use these partitions to separately discretise the near-field, boundary layer region and the rest of the domain. In the former, a structured prismatic or hexahedral mesh can be generated by using geometric information obtained from the medial axis. In the latter, traditional unstructured mesh generation techniques can be used. A restriction of such partitioning to the near-wall regions and an appropriate design of the medial object partitioning reduces significantly the complexity of the generation process and makes it possible to obtain high-quality boundary layer type hybrid meshes near the wall surfaces.

The generation of a high-order mesh using this semi-structured approach involves two steps. First a straight-sided mesh is generated with a coarse boundary layer mesh composed of a single layer within the medial object based partitions adjacent to the wall boundaries. Additional points are then added, following essentially the method described in reference [70], to obtain a high-order curvilinear mesh compliant with the CAD[3] definition. Next a boundary layer mesh is generated using the isoparametric approach proposed in reference [54] where elements in the coarse mesh adjacent to the wall are subdivided along the normal direction, according to a user-defined resolution. In this work, an extension to this method is described that leverages medial axis decomposition to generate high-quality meshes in corners and junctions by performing a split in two separate directions normal to each surface of the corner section, similarly to what is done in multi-normal advancing layer techniques [6]. This approach is flexible and modular, and allows a variety of resolutions to be defined from a base coarse high-order mesh that remains unchanged.

1.4 Mesh adaptation

Following the generation of a valid high-order mesh and the simulation of the desired flow configuration, a mesh adaptation strategy can finally be employed. Historically, research on adaptation has focused on remeshing as a means to obtain a mesh better suited for the simulation at hand [27, 88]. While a mesh can be obtained that exactly fits a certain metric field based on error indicators, the solution must be interpolated onto the new mesh and the simulation restarted. Instead, one can look at h-adaptation [40] where only local mesh operations are performed, such as element splitting, edge swapping or node insertion. These fast operations leave the rest of the mesh unchanged and accelerate the iterative process. The set of allowable operations can also be limited to simple vertex movement, in which case moving meshes or r-adaptation is obtained [12, 31]. The mesh connectivities, in this case, remain unchanged. The number of DOF also remains constant, which can be considered problematic when more resolution is required than the mesh as a whole can offer. Finally, in more recent years, high-order methods have brought the possibility to achieve p-adaptation [40, 24]. In this case, the mesh remains entirely unchanged

[3] CAD: computer-aid design.

and only the local polynomial order used by the solver is adapted to provide more or less resolution to certain regions. In this thesis, a proof-of-concept strategy is proposed, based on *rp*-adaptation. Both methods conserve mesh connectivities. *r*-adaptation is used in regions where *p*-adaptation is inefficient, as explained below, whereas *p*-adaptation is able to increase or decrease the total number of DOF.

For the *r*-adaptation procedure, a variational optimiser is used to deform the mesh. This variational framework was originally designed to optimise high-order meshes, in which a functional of the deformation energy is minimised [80]. This framework can be modified to target an element shape and size within the mesh. By targeting a small element size in regions of shocks, the optimiser deforms the mesh and clusters nodes in said regions. This concept of moving mesh has been used on linear meshes [12, 31] and this thesis offers insight on its application to high-order meshes. By effectively redistributing DOF, *h*-type refinement is obtained at flow discontinuities [1, 59]. A *p*-adaptation procedure proposed in [24] is then applied to this *r*-adapted mesh to better resolve regions of smooth flow.

For both adaptation aspects, a discontinuity sensor [60] is used that is easily computed and essentially looks at the energy of the higher modes to determine the level of resolution of the solution. The purposes of this sensor are three-fold: first it adds artificial viscosity to the governing equations, based on values of the sensor, to stabilise the solution in the presence of shocks; second it identifies regions of flow discontinuities based on values of the sensor, as used for the artificial viscosity, to focus *r*-adaptation; and third it locally increases or decreases the local order of the polynomial approximation based on the values of the sensor.

1.5 Outline

This section concludes Ch. 1, which gives an overview of the current challenges in the field of high-order mesh generation and adaptation as well as describes the objectives of this thesis and the associated methodologies to achieve them. An outline of the thesis is presented below.

Ch. 2 recalls the spectral/*hp* element formulation [35, 17, 50] used throughout this work for the discretisation of domains and PDE[4]. Sec. 2.2, in particular, reviews the inviscid compressible flow equations and discretises them. Finally, Sec. 2.2.3 gives a description of the discontinuity sensor used later in this thesis for compressible flow simulations and mesh adaptation. This version of the sensor was improved by G. Castiglioni in [44].

[4] PDE: partial differential equation.

Ch. 3 covers the proposed generation of naturally curved quadrilateral meshes using a guiding field approach, a collaboration with D.A. Kopriva, co-author of [45, 46]. The underlying mathematical formulation is first introduced in Sec. 3.1 before the practical procedure is explained in Sec. 3.2. Sec. 3.3 goes on to demonstrate the benefits of the methods through various examples.

Ch. 4 continues on the topic of high-order mesh generation, in the context of three-dimensional domains, using the proposed hybrid approach. This work is a collaboration with the International TechneGroup, co-authors of [49, 47]. Sec. 4.1 first describes the CAD interface that handles the geometrical queries during the generation of both linear and high-order meshes. Sec. 4.2 then provides an overview of the medial object approach and discusses its application to the decomposition of the domain into near-field and far-field regions. Sec. 4.1 and Sec. 4.2 cover the work of the International TechneGroup and are included for completeness. The generation of a high-order mesh from this linear mesh is described in Sec. 4.3. Sec. 4.4 presents meshes generated by this methodology for two geometries. A more recent extension of this method is finally presented in Sec. 4.5 to more specifically handle topologies typically associated with streamlined bodies.

Ch. 5 switches the focus onto mesh adaptation and covers the variational framework used for r-adaptation. Sec. 5.1 first recalls the formulation of the variational approach to high-order mesh optimisation [80], which is then modified to achieve r-adaptation in Sec. 5.2. The latter section includes analytical examples while Sec. 5.3 provides evidence that it can be driven by *a posteriori* error estimates, resulting from a collaboration with F. Naddei.

Ch. 6 then aims at using this new r-adaptation technique in combination with p-adaptation. While Sec. 6.3 presents the general workflow of this dual approach, Sec. 6.1 and Sec. 6.2 describe the specific uses of r- and p-adaptation [24], respectively. Finally two numerical examples are presented in Sec. 6.4 in vastly different flow conditions.

This thesis concludes with Ch. 7, which summarises the work covered and gives insight into future work.

1.6 Publications

The work associated with this thesis has resulted in multiple publications. These are listed below in link with their associated chapter in this document. This list includes some papers currently under review.

Ch. 3 on two-dimensional quadrilateral mesh generation:

- [45] J. Marcon, D.A. Kopriva, S.J. Sherwin, and J. Peiró. A high resolution PDE approach to quadrilateral mesh generation. *Journal of Computational Physics*, 399C:108918, 2019

- [46] J. Marcon, D.A. Kopriva, S.J. Sherwin, and J. Peiró. Naturally curved quadrilateral mesh generation using an adaptive spectral element solver. In *Proceedings of the 28th International Meshing Roundtable*, Buffalo, NY, USA, 2019. in press

Ch. 4 on three-dimensional hybrid mesh generation:

- [49] J. Marcon, M. Turner, J. Peiró, D. Moxey, C. Pollard, H. Bucklow, and M.R. Gammon. High-order curvilinear hybrid mesh generation for CFD simulations. In *2018 AIAA Aerospace Sciences Meeting*, Reston, Virginia, 2018. American Institute of Aeronautics and Astronautics.

- [47] J. Marcon, J. Peiró, D. Moxey, N. Bergemann, H. Bucklow, and M.R. Gammon. A semi-structured approach to curvilinear mesh generation around streamlined bodies. In *AIAA Scitech 2019 Forum*, Reston, Virginia, 2019. American Institute of Aeronautics and Astronautics.

Ch. 5 on *r*-adaptation, excluding Sec. 5.3:

- [48] J. Marcon, M. Turner, D. Moxey, S.J. Sherwin, and J. Peiró. A variational approach to high-order r-adaptation. In *26th International Meshing Roundtable*, 2017.

Ch. 6 on *rp*-adaptation and Sec. 2.2.3 on the shock sensor:

- [44] J. Marcon, G. Castiglioni, D. Moxey, S.J. Sherwin, and J. Peiró. *rp*-adaptation for compressible flows, 2019, arXiv:physics.comp-ph/1909.10973. under review.

Other authored papers whose content is not directly covered in this thesis:

- [20] J. Cohen, J. Marcon, M. Turner, C.D. Cantwell, S.J. Sherwin, J. Peiró, and D. Moxey. Simplifying high-order mesh generation for computational scientists. In M. Atkinson and S. Gesing, editors, *Proceedings of the 10th International Workshop on Science Gateways*, Edinburgh, Scotland, 2019. CEUR-WS.org.

- [50] D. Moxey, C.D. Cantwell, Y. Bao, A. Cassinelli, G. Castiglioni, S. Chun, E. Juda, E. Kazemi, K. Lackhove, J. Marcon, G. Mengaldo, D. Serson, M. Turner, H. Xu, J. Peiró, R.M. Kirby, and S.J. Sherwin. Nektar++: Enhancing the capability and application of high-fidelity spectral/*hp* element methods. *Computer Physics Communications*, 107110, 2019. in press.

Chapter 2

Spectral/*hp* element methods

In this work, the spectral/*hp* element methods formulation, described in detail in reference [35] and implemented in *Nektar++* [17, 50], is used. The fundamentals of the methods are briefly described in what follows. These methods have not been implemented as part of this thesis and are only included for completeness. The interested reader may refer to the references above for a more detailed description and analysis.

Sec. 2.1 first summarises the general formulation of the spectral/*hp* element methods. Sec. 2.2 then recalls the inviscid compressible flow equations and discretises them. Importantly, Sec. 2.2.3 describes a discontinuity sensor used to stabilise the compressible flow simulations and, later, for mesh adaptation.

2.1 General formulation

The numerical solution of PDE of the form $\mathcal{L}u = 0$ over a domain Ω is considered. The domain Ω is taken as a set of finite elements, Ω_e — the mesh — such that $\Omega = \cup\Omega_e$ and $\Omega_{e_1} \cap \Omega_{e_2} = \partial\Omega_{e_1 e_2}$ is either an empty set or the interface between two elements and is of one dimension less than the mesh. The PDE problem is solved in the weak sense and requires that $u|_{\Omega_e}$ is in the Sobolev space $H^k(\Omega_e)$, $k = 0, 1$. In the CG[1] formulation, the solution is required to be in H^1; in DG[2], H^0.

The solution to $\mathcal{L}u = 0$ is formulated in weak form: *find* $\vec{v} \in H^k(\Omega)$ *such that*

$$a(\vec{v}, \vec{w}) = l(\vec{w}) \quad \forall \vec{w} \in H^k(\Omega), \tag{2.1}$$

where $a(\cdot, \cdot)$ is a symmetric bilinear form, $l(\cdot)$ is a linear form, and $H^k(\Omega)$ is formally

[1] CG: continuous Galerkin.

[2] DG: discontinuous Galerkin

defined as

$$H^k(\Omega) = \{\vec{w} \in L^2(\Omega) \mid D^\alpha \vec{w} \in L^2(\Omega) \; \forall \; |\alpha| \leq k\}. \tag{2.2}$$

This problem is solved numerically and therefore solutions are considered in a finite dimensional subspace $V_N \subset H^k(\Omega)$. The problem is: *find* $\vec{v}^\delta \in V_N$ *such that*

$$a(\vec{v}^\delta, \vec{w}^\delta) = l(\vec{w}^\delta) \quad \forall \vec{w}^\delta \in V_N, \tag{2.3}$$

augmented with appropriate BC. In the CG formulation, the condition $V_N \subset C^0$ is also enforced.

A weighted sum of N trial functions $\Phi_n(\vec{x})$ defined on Ω is used so $\vec{v}^\delta(\vec{x}) = \sum_n \hat{v}_n \Phi_n(\vec{x})$. This transforms the problem to that of finding the coefficients $\hat{u}_n$ that define $\vec{v}^\delta(\vec{x})$ within an element. To obtain a unique choice of coefficients $\hat{u}_n$, a restriction on $R = \mathcal{L}\vec{v}^\delta$ is placed that its L^2 inner product, with respect to the test functions $\Psi_n(\vec{x})$, is zero. In the Galerkin projection one chooses the test functions to be the same as the trial functions, i.e. $\Psi_n = \Phi_n$.

The contributions of each element in the domain must be taken into account to construct the global basis Φ_n. A parametric mapping $\mathcal{X}_e : \mathcal{E} \to \Omega_e$ exists from a standard reference element $\mathcal{E} \subset [-1,1]^d$ to each element Ω_e. This mapping is given by $\vec{x} = \vec{\mathcal{X}}_e\left(\vec{\xi}\right)$. It is important to distinguish $\vec{x}$, the physical coordinates, from the $\vec{\xi}$, the coordinates in the reference space.

A local polynomial basis is constructed on the reference element to represent the solution. A one-dimensional order-P basis is a set of polynomials $\Phi_p(\xi), 0 \leq p \leq P$, defined on the reference segment $-1 \leq \xi_1 \leq 1$. In two- and three-dimensional reference regions, a tensor basis is used, where the polynomial space is constructed as the tensor product of one-dimensional bases on segments, quadrilaterals and hexahedral reference regions. In *Nektar++*, triangular, tetrahedral, prismatic and pyramidal elements are created by collapsing one or more of the coordinate directions to create singular vertices. This allows it to support, for this work, easy-to-generate triangular meshes.

Finally the discrete solution in a physical element Ω_e can be expressed as

$$\vec{v}^\delta(\vec{x}) = \sum_n \hat{v}_n \phi_n(\mathcal{X}_e^{-1}(\vec{x})), \tag{2.4}$$

with $\hat{v}_n$ the coefficients computed by the Galerkin procedure. The solution space is restricted to

$$V := \{\vec{v} \in H^k(\Omega) \mid \vec{v}|_{\Omega_e} \in \mathcal{P}_P(\Omega_e)\}, \tag{2.5}$$

where $\mathcal{P}_P(\Omega_e)$ is the space of order-P polynomials on Ω_e.

An assembly operator can then be designed to assemble the element contributions to the global solution. In the CG formulation, elemental contributions of neighbours are summed to enforce C^0-continuity. In the DG formulation, such mappings transfer flux values from the element interfaces into the global solution vector.

2.1.1 High-order meshes

High-order meshes often require different measures of quality than linear meshes. The distortion of curvilinear elements is typically a source of issues for high-order solvers [78]. As seen above, elements in a high-order curvilinear mesh are represented by a parametric mapping $\mathcal{X}_e : \mathcal{E} \to \Omega_e$. The quality of such mapping can often be assessed by a metric called the scaled Jacobian J_s [28]:

$$J_s = \frac{\min_{\xi \in \Omega_{st}} J(\xi)}{\max_{\xi \in \Omega_{st}} J(\xi)}. \tag{2.6}$$

For a more thorough analysis of this metric and a comparison with other measures of quality, see reference [78].

The scaled Jacobian, in essence, looks at the level of deformation of the high-order element. It is the ratio between the minimum and the maximum value of the Jacobian of the mapping over the element. For a straight-sided simplex, $J_s = 1$. A negative scaled Jacobian, on the other hand, indicates that the element is invalid because of self-intersection. While distortion is often unavoidable at boundaries, the objective is often to reduce distortion as much as possible and bring the scaled Jacobian to a value close to 1.

2.2 Compressible flows

In this work, no viscous effects are considered. Boundary layers add another level of difficulty which is beyond the scope of this thesis. The Euler equations of compressible flow are therefore used. They are, notably, stabilised by the use of a dissipative flux, added in the presence of shocks, with respect to a discontinuity sensor.

2.2.1 Governing equations

The Euler equations of inviscid compressible flow are written, in a two-dimensional Cartesian frame of reference with coordinates $\boldsymbol{x} = (x_1, x_2)$ within a domain Ω with boundary Γ, as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \left[\, \mathbf{F}_c(\mathbf{u}) + \mathbf{F}_d(\mathbf{u}, \nabla \mathbf{u}) \,\right] = \mathbf{0}. \tag{2.7}$$

Here $\mathbf{u} = [\rho, \rho v_1, \rho v_2, \rho E]^t$ is the vector of conserved variables, where ρ is the density, the Cartesian components of the velocity are $\boldsymbol{v} = (v_1, v_2)$, and E is the total energy. The terms $\mathbf{F}_c$ and $\mathbf{F}_d$ denote the convective and dissipative fluxes, respectively. A dissipative flux is required to stabilise the solution in the presence of shocks which is chosen to be of the form

$$\mathbf{F}_d = -\mu_a(\mathbf{u}) \nabla \mathbf{u}, \tag{2.8}$$

where μ_a is an artificial viscosity coefficient, discussed in detail below. The components of the convective flux $\mathbf{F}_c = (\mathbf{f}_1, \mathbf{f}_2)$ are given by

$$\mathbf{f}_1 = \begin{Bmatrix} \rho v_1 \\ P + \rho v_1^2 \\ \rho v_1 v_2 \\ \rho v_1 H \end{Bmatrix}, \quad \mathbf{f}_2 = \begin{Bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ P + \rho v_2^2 \\ \rho v_2 H \end{Bmatrix}, \tag{2.9}$$

where H is the total enthalpy and P is the pressure. The total enthalpy is defined as

$$H = E + \frac{P}{\rho} \tag{2.10}$$

and, to close the system, the pressure for a perfect gas is given by

$$P = (\gamma - 1)\rho \left(E - \frac{v_1^2 + v_2^2}{2} \right), \tag{2.11}$$

where γ is the ratio of specific heats and its value for air is $\gamma = 1.4$.

The setting of the problem is completed through a suitable choice of initial and boundary conditions. Only steady-state problems are of interest in this thesis and therefore all simulations are started with a uniform flow at the given free-stream Mach number and flow incidence. Solid walls are modelled through the no-flow condition, $\boldsymbol{v} \cdot \boldsymbol{n} = 0$, where $\boldsymbol{n}$ denotes the wall outer normal. Far-field boundaries are weakly imposed through the normal boundary fluxes by specifying free-stream conditions, $\mathbf{u} = \mathbf{u}_\infty$, outside the boundary and evaluating the normal fluxes through a Riemann solver that accounts for the propagation of information across the boundary.

2.2.2 Discretisation

In this thesis, a discrete solution of Eq. (2.7) is found via a high-order spectral/*hp* DG discretisation described in Sec. 2.1. The computational domain Ω is subdivided into N_{el} non-overlapping elements. A mixed formulation [5] is used and Eq. (2.7) writes as

$$\mathbf{g} - \nabla \mathbf{u} = \mathbf{0}\,, \tag{2.12}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot [\, \mathbf{F}_c(\mathbf{u}) + \mathbf{F}_d(\mathbf{u}, \mathbf{g}) \,] = \mathbf{0}. \tag{2.13}$$

A discrete approximation within an element Ω^e is sought of the form (2.4). In the DG form, both the solution and the test functions are discontinuous at the interface between elements.

Following the standard Galerkin procedure, a weak form of the mixed formulation of Eqs. (2.12)–(2.13) is obtained as follows. The discrete version of Eq. (2.12) reads

$$\sum_{e=1}^{N_{el}} \int_{\Omega_e} v_i^e \left(\mathbf{g}_h^e - \nabla \mathbf{u}_h^e \right) d\Omega^e = \mathbf{0}\,; \quad i = 1, \ldots, N, \tag{2.14}$$

where $\mathbf{g}_h^e$ and $\mathbf{u}_h^e$ are the discrete approximations of the solution within element Ω^e. Using an approximation of the form (2.4) for both $\mathbf{u}_h^e$ and $\mathbf{g}_h^e$, and applying Gauss' theorem, Eq. (2.14) becomes

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(v_i^e \sum_{j=1}^{N} \mathbf{g}_j^e v_j^e + \nabla v_i^e \sum_{j=1}^{N} \mathbf{u}_j^e v_j^e \right) d\Omega^e - \sum_{e=1}^{N_{el}} \int_{\Gamma^e} v_i^e \left(\sum_{j=1}^{N} \mathbf{u}_j^e v_j^e \right) \boldsymbol{n}\, d\Gamma^e = \mathbf{0}\,; \quad i = 1, \ldots, N, \tag{2.15}$$

where Γ^e denotes the boundary faces of element Ω^e. The solution of Eq. (2.15) gives the discrete values of the first-order derivatives $\mathbf{g}_h^e$.

The weak form of Eq. (2.13) is obtained in a similar fashion to give

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} v_i^e \sum_{j=1}^{N} \frac{d\mathbf{u}_j^e}{dt} v_j^e \, d\Omega - \sum_{e=1}^{N_{el}} \int_{\Omega^e} \nabla v_i^e \sum_{j=1}^{N} \mathbf{F}_j^e v_j^e \, d\Omega + \sum_{e=1}^{N_{el}} \int_{\Gamma_e} v_i^e \sum_{j=1}^{N} (\mathbf{F}_j^e \cdot \boldsymbol{n}) v_j^e \, d\Gamma = 0\,; \quad i = 1, \ldots, N. \tag{2.16}$$

The solution is discontinuous at the interface between elements and the integrand in the boundary integral of Eq. (2.16) is substituted by a numerical flux function. The convective normal flux at an interface is approximated by a numerical flux calculated via a Riemann solver

$$[(\mathbf{F}_c)_i^e \cdot \boldsymbol{n}]_{\Gamma_e} \approx \mathcal{H}^c(\mathbf{u}_e, \mathbf{u}_{e^+}; \boldsymbol{n}), \tag{2.17}$$

where $\mathbf{u}_{e^+}$ and $\mathbf{u}_e$ are the values of the conservative variables on the external and internal sides of the interface with respect to the e^{th} element. This mechanism allows information to pass from one element to the other. The evaluation of the diffusive normal flux at the interface follows the LDG[3] formulation [19], where it is approximated by

$$[(\mathbf{F}_d)_i^e]_{\Gamma_e} = \{\{\mathbf{F}_d\}\} + \mathbf{C}_{12}[\![\mathbf{F}_d]\!] + C_{11}[\![\mathbf{u}]\!] \tag{2.18}$$

and, similarly,

$$[(\mathbf{u})_i^e]_{\Gamma_e} = \{\{\mathbf{u}\}\} - \mathbf{C}_{12}[\![\mathbf{u}]\!], \tag{2.19}$$

where $\mathbf{C}_{12} = \frac{1}{2}\mathbf{n}$, and C_{11} is an order 1 constant. The average and jump operators are defined as

$$\{\{u\}\} = \frac{1}{2}\left(u^+ + u^-\right), \qquad [\![u]\!] = \left(u^+\boldsymbol{n}^+ + u^-\boldsymbol{n}^-\right), \tag{2.20}$$

$$\{\{\mathbf{u}\}\} = \frac{1}{2}\left(\mathbf{u}^+ + \mathbf{u}^-\right), \qquad [\![\mathbf{u}]\!] = \left(\mathbf{u}^+ \cdot \boldsymbol{n}^+ + \mathbf{u}^- \cdot \boldsymbol{n}^-\right). \tag{2.21}$$

2.2.3 Shock capturing

This DG discretisation of the Euler equations requires the addition of the diffusion flux $\mathbf{F}_d$ to stabilise the solutions in the presence of shock waves. The term μ_a in Eq. (2.8) is an artificial viscosity coefficient that allows dissipation to be selectively applied to shocks. For consistency $\mu_a \sim h/p$, the following formulation [4] is used

$$\mu_a \sim \frac{h}{p}\lambda_{\max}, \tag{2.22}$$

where $\lambda_{\max} = |u| + c$ is the local maximum wave speed of the system. The characteristic cell length h is chosen as the minimum edge length of an element. Finally, for the artificial viscosity to vanish outside shocks, it needs to be proportional to a shock sensor S such that

$$\mu_a = \mu_0 \frac{h}{p}\lambda_{\max} S, \tag{2.23}$$

where $\mu_0 = O(1)$ is a constant. To build the shock sensor, a modal resolution based indicator [60] is used, which is element-wise constant and defined via an intermediary term

$$s_e = \log_{10}\left(\frac{\langle q - \tilde{q}, q - \tilde{q}\rangle}{\langle q, q\rangle}\right), \tag{2.24}$$

[3] LDG: local discontinuous Galerkin.

where $\langle \cdot, \cdot \rangle$ represents an L^2 inner product, q and $\tilde{q}$ are the full and truncated expansions of a state variable

$$q(x) = \sum_{i=1}^{N(p)} \hat{q}_i \phi_i, \quad \tilde{q}(x) = \sum_{i=1}^{N(p-1)} \hat{q}_i \phi_i, \tag{2.25}$$

where ϕ_i are the basis functions, $\hat{q}_i$ the associated coefficients, and $N(p)$ the size of the expansion of order p. In this case, the test variable q is chosen to be the density ρ because it is readily available to the solver and it is better suited to capture a wide range of flow features. To spatially smooth out the variation of the values of the sensor, the constant element-wise sensor is computed as follows

$$S = \begin{cases} 0, & s_e \leq s_0 - \kappa, \\ \frac{1}{2}\left(1 + \sin \frac{\pi(s_e - s_0)}{2\kappa}\right), & |s_e - s_0| \leq \kappa, \\ 1, & s_e \geq s_0 + \kappa, \end{cases} \tag{2.26}$$

with $s_0 \sim \log_{10}(p^4)$ from an analogy to Fourier coefficients decaying as $1/p^2$, and κ needs to be sufficiently large to obtain a smooth shock profile. s_0 is chosen as

$$s_0 = -s_\kappa - 4.25 \log_{10}(p), \tag{2.27}$$

where s_κ and κ can be adjusted for a specific problem.

Chapter 3

Two-dimensional quadrilateral mesh generation

An approach is presented in this chapter to generate quadrilateral block decompositions, and subsequently quadrilateral meshes, of arbitrary two-dimensional domains. Quadrilateral block decompositions and meshes offer significant advantages for numerical methods over triangular meshes. In the spectral/*hp* element methods [35], two- and three-dimensional elements are constructed by tensor product of one-dimensional polynomial bases, producing quadrilaterals and hexahedra respectively. Other types of high-dimensional elements, e.g. triangles and tetrahedra, are created by collapsing one or more of the coordinate directions to create singular edges and vertices. In addition, simplexes, i.e. triangles and tetrahedra, are naturally stiffer than tensor product elements, i.e. quadrilaterals and hexahedra. For these reasons, some solvers are developed without support for simplex elements.

Several approaches to unstructured quadrilateral mesh generation have become popular over the years [8, 3]. A first trivial approach consists of tesselating the domain then recombining triangles to form quadrilaterals. In two dimensions, an all-quadrilateral mesh can be obtained if there is an even number of triangles but such a guarantee does not exist in three dimensions. This approach also yields elements that are often distorted. The level of distortion can be reduced by using, for example, a Centroidal Voronoi tesselation that can optimise for a measure of the quality using an L_p-norm, resulting in mostly square Voronoi cells [22, 23]. Another popular approach is based on octrees [68], where a fully cartesian grid is first generated that fully covers the domain without conforming to boundaries. This mesh is then localy refined and trimmed at the boundaries. It is finally snapped onto the boundaries to conform to them. Although the interior mesh is entirely regular, the last step tends to produce deformed elements at the boundaries. While the octree based approach goes inside out, another set of techniques, often called paving, advancing fronts or yet advancing layers, takes an outside in approach [6]. These typically take a boundary discretisation and march inward by offsetting boundary elements, thus creating quadrilaterals in two dimensions, hexahedra in three dimensions. Extra care must be taken when these advancing fronts collide, often yielding distorted elements. More recently, research on the medial axis [3] has looked at analysing shapes to extract geometrical information and finding irregular nodes in the domain. This technology is able to generate more regular meshes that are often block structured.

This chapter looks at another set of approaches, based on guiding fields [8], that are able to decompose domains into quadrilateral blocks. The method presented here is based on the concept of crosses, two-dimensional objects that are invariant to rotations of $\pi/2$. Importantly they are aligned with boundaries and give a guiding direction for the generation of block decompositions. From this notion of crosses, a guiding field is formulated, which can be solved with a Laplace solver. In this work, a high-order spectral/*hp* element solver is used but it is important to note that a low-order solver would also be appropriate. Finally, this guiding field, whose formulation is described in Sec. 3.1, creates valid quadrilateral block decompositions.

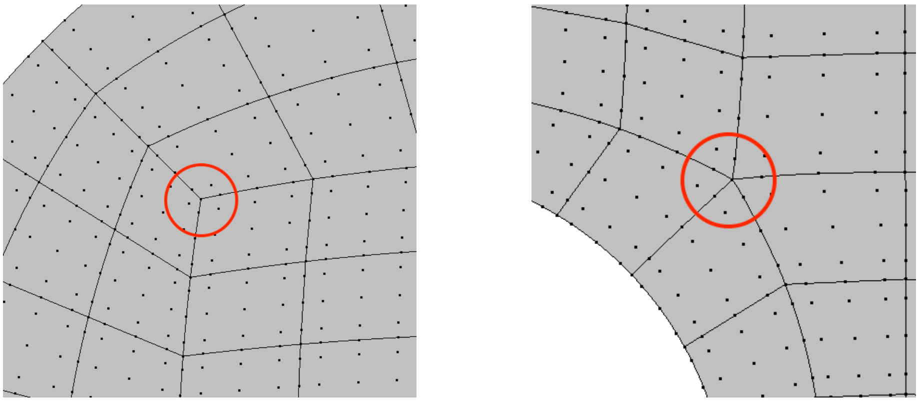
An important contribution of this chapter concerns the mesh generation aspect, described in Sec. 3.2. These curved blocks, obtained from the guiding field, in turn yield *a priori* curved, high-order meshes. These are naturally valid and various examples demonstrate that aspect in Sec. 3.3.

3.1 Formulation

The problem is to subdivide any two dimensional domain, Ω, that has piecewise smooth boundaries into quadrilateral subdomains. The domain might be simply or multiply connected. The decomposition will be regular if all the corners of the quadrilaterals have valence four, that is, each corner node connects four edges. In general, it will not be possible to generate a purely regular mesh, especially in multiply connected domains. Instead, some of the nodes will be *irregular*, where the valence will be larger or smaller than four, like those observed in Fig. 3.1.

3.1.1 Cross field

To motivate the high-resolution approach to the quadrilateral decomposition of a domain, Ω, and to put it into context, the notion of a cross is recalled below. A



(a) Valence three.

(b) Valence five.

Figure 3.1: High-order quadrilateral meshes: irregular nodes encircled in red, interior high-order nodes represented by dots.

cross can be represented by

$$C\left(\psi\right)=\left\{\vec{c}_k=\left(c_x,c_y\right)_k=\left(\cos\left(\psi+k\frac{\pi}{2}\right),\sin\left(\psi+k\frac{\pi}{2}\right)\right)\right\}, \tag{3.1}$$

for $k=0,1,2,3$. Here, the tangent angle, or phase, ψ, is computed from the four quadrant inverse tangent

$$\psi=\frac{1}{4}\text{atan2}\left(v,u\right)\in\left[-\frac{\pi}{4},\frac{\pi}{4}\right] \tag{3.2}$$

so that a cross is represented by four unit vectors $\vec{c}_k$ at any point $\vec{x}\in\Omega$ given a vector field $\vec{v}=\left(u\left(\vec{x}\right),v\left(\vec{x}\right)\right)$ at which $\vec{v}\neq 0$ from the principal direction. See Fig. 3.2. The important property of a cross is its 4-way rotational symmetry. Rotation by an angle $\pi/2$ does not change the cross. The axes of the cross therefore lie in the tangent and normal directions of the guiding field lines.

The function ψ is tangent or orthogonal (due to the jump in the atan2 function) to the streamlines of the field $\vec{v}$. Hence, it is parallel to one of the branches of a cross. It is undefined at $\vec{v}=(0,0)$. Since ψ is computed from the arctangent, there will be a jump of of value $\pi/2$ in ψ depending on the signs of u and v even if $\vec{v}$ is smooth. Lines in the field across which ψ jumps are called *jump lines*.

A full discussion of cross fields and methods based on them is beyond the scope of this thesis. See, for example, references [13, 14, 84, 83] for more comprehensive discussions.

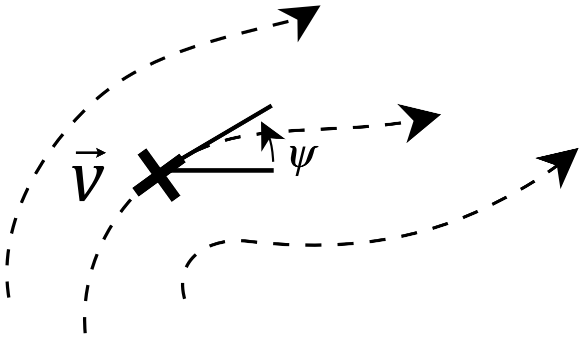


Figure 3.2: A cross defined in a guiding field $\vec{v}$.

Guiding field

It is necessary, then, to find the guiding field $\vec{v}$ from which to find the irregular nodes and trace separatrices. The only values that can be set *a priori* are on the boundary $\partial\Omega$ of Ω. The field is aligned with the boundary to ensure that the mesh is aligned there. Therefore,

$$\vec{v}_b = (u_b, v_b) = (\cos(4\theta_b), \sin(4\theta_b)), \tag{3.3}$$

where θ_b is the tangent angle of the boundary where the field is being computed. The vector $\vec{v}_b$ at any point along the boundary defines a cross (3.1) at that point. The factor of four in Eq. (3.3) ensures the same u, v values for each 90° rotation of the angle, and hence the rotational symmetry. Fig. 3.3 shows example crosses on the boundary of a quarter circle domain.

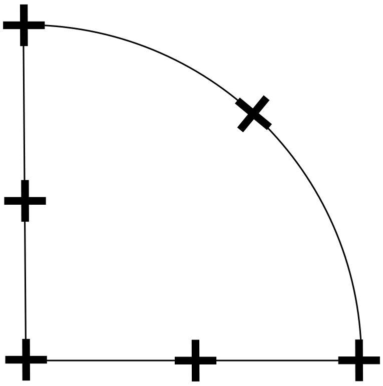


Figure 3.3: Boundary crosses on the quarter circle.

The following three observations can be made from Eq. (3.3) about the boundary field $\vec{v}_b$:

- Along any circle, there are eight zeros each for u and v, and those zeros are not coincident.

- Along smooth portions of the boundary curves and at corners whose angle is a multiple of $\pi/2$, u and v are continuous.

- At boundary corners with angles not divisible by $\pi/2$ there is a jump discontinuity in $\vec{v}_b$ and the cross field.

Once the boundary values are specified using expression (3.3), they are smoothly propagated to the interior. Smoothing procedures that have been used in the past include marching the associated crosses to the interior [25], minimizing an energy functional [83], or solving a Laplace problem [64]. Solving a Laplace problem for u and v

$$\begin{cases} \nabla^2 \vec{v} = 0, & \vec{x} \in \Omega \\ \vec{v} = \vec{v}_b, & \vec{x} \in \partial\Omega \end{cases} \tag{3.4}$$

guarantees that the field $\vec{v}$ is smooth in the interior of the domain and satisfies the maximum principle. The solution of the Laplace problem is equivalent to the minimisation of the Dirichlet energy. Eqs. (3.4) makes it possible to use a high-resolution approach such as the spectral element methods. In particular, the use of a DG approximation allows geometries where the corner angles generate jumps in the guiding field to be handled.

CG and DG spectral/*hp* element methods

A key difference between this work here and that of previous work is that high-resolution solutions are computed for the BVP[1] (3.4) with either a CG or a DG spectral/*hp* element method on a triangular mesh [35]. Spectral/*hp* element methods are spectrally accurate, meaning that the convergence rate depends only on the smoothness of the solution. They are high-resolution in that they use a large number of DOF within an element. Unlike traditional finite element methods, the high-order polynomial expansion of the solution inside each element permits to locate irregular nodes, identify their valence, and finally trace separatrices with better accuracy. The formulations of the CG and DG spectral/*hp* element methods were described in Sec. 2.1.

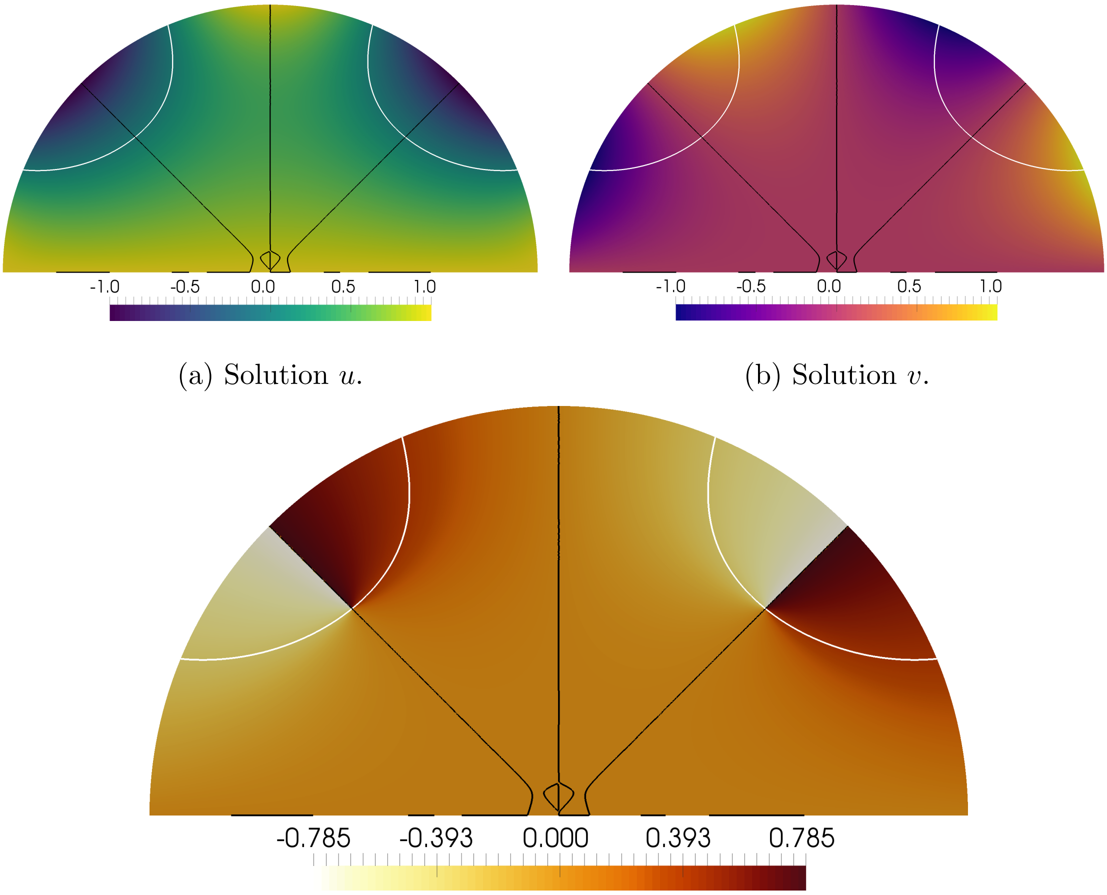
3.1.2 Critical points

Points where $v = u = 0$ are called *critical points* in the guiding field and *singular points* in the cross field. They are points where ψ cannot be determined uniquely.

[1] BVP: boundary value problem.

Such points become the interior *irregular nodes* (valence $\neq 4$) in the quadrilateral decomposition.

The form of the boundary values usually implies that critical points will exist in the interior of the domain. Since there are interlaced zeros in u and v on the boundary, such as occur when the boundary is smooth and θ_b varies continuously, the fact that the interior field is smooth implies that zero contours of u and v must cross somewhere in the interior, thereby creating a critical point. Zeros in u will exist along the boundary at any point where θ_b smoothly passes through $\theta_b = k\pi/8$, k odd, and in v for k even. Therefore, one can infer the existence of interior irregular nodes in the final quadrilateral mesh from the boundary curves. A semi-circular section of a boundary, for instance will have four zeros in u_b and five in v_b and, in the absence of other nearby features, will create two interior critical points. See Fig. 3.4. Although the cross field directions are undefined at critical points, the field $\vec{v}$ is smooth because the solutions of the Laplace problem (3.4) are regular.



(a) Solution u.

(b) Solution v.

(c) Computed ψ.

Figure 3.4: Guiding field on the half disc: isocontours of $u = 0$ and $v = 0$ shown in white and black respectively.

Irregular nodes in the mesh occur as recognisable critical point topologies in the $\vec{v}$ field [77, 13, 14]. They can be categorised analytically by the Poincaré index

$$i_\gamma = \frac{1}{2\pi} \oint_\gamma d\phi, \tag{3.5}$$

where

$$\phi = \text{atan2}\,(v, u) = 4\psi. \tag{3.6}$$

When γ encloses a critical point, $i_\gamma = \pm 1$, where -1 corresponds to a saddle point. Other critical points have index $+1$. If there is no critical point inside the contour γ then the index is zero. The categorisation of the critical point lets one determine the number of streamlines to emanate from the irregular node, and hence its valence in the mesh. When $i_\gamma = -1$ there are five separatices. When $i_\gamma = +1$ there are three [37], [83, Lemma 5.1]. It is possible to have an index of $+2$ at which six elements would meet. However such points are unstable and in practice will split into separate critical points of lower valence. For this reason, such critical points are not considered here.

Using definition (3.5) and relation (3.6), the following integral is similarly defined

$$I_c = \frac{1}{\pi/2} \oint_c \frac{d\psi}{d\theta} d\theta, \tag{3.7}$$

where c is a (small) counter-clockwise circular contour centred on the critical point. Table 3.1 shows the values of I_c for three separatrix topologies in the neighbourhood of a critical point. It is equal to the sum of the jumps in ψ around the contour. The *valence* $\mathcal{V}$ of the critical point is then

$$\mathcal{V} = 4 - I_c. \tag{3.8}$$

Note, for consistency, that if there is no critical point inside the contour then $I_c = 0$ and the point is regular with valence four.

Table 3.1: Value of I_c, from Eq. (3.7), and the associated valence.

I_c	Valence ($\mathcal{V}$)
-1	5
0	4
$+1$	3

If one generalises the integral I_c to be over a part of the circle, one can characterise boundary vertices, too [14]. Let

$$I\left(\theta_0, \theta_f\right) = \frac{1}{\pi/2} \int_{\theta_0}^{\theta_f} \frac{d\psi}{d\theta} d\theta, \tag{3.9}$$

so that $I_c = I\left(0, 2\pi\right)$. At a boundary point, then, the number of quadrilaterals attached to the point (the boundary itself already being a streamline by construction) is

$$\mathcal{V} = \frac{\Delta\theta}{\pi/2} - I\left(\theta_0, \theta_f\right), \tag{3.10}$$

where $\Delta\theta = \theta_f - \theta_0$. $I\left(\theta_0, \theta_f\right)$ can be viewed as a correcting factor to $\Delta\theta$ towards a multiple of $\frac{\pi}{2}$.

3.1.3 Streamlines

The guiding field is also used to trace *streamlines* that will form the *separatrices* of the block decomposition. The problem is formulated as finding the trajectory $\vec{x}(t)$ that satisfies

$$\frac{d\vec{x}}{dt} = \tilde{v}\left(\psi'(\vec{x})\right), \tag{3.11}$$

where t is the integration parameter and $\tilde{v}\left(\psi'(\vec{x})\right)$ is the adjusted guiding field vector. Eq. (3.2) gives a guiding field $\psi \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$, but crosses are invariant to rotations of $\pi/2$. It is therefore necessary to account for the branches of the cross that don't lie within $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. These adjusted guiding directions can be obtained by adding a certain number $k = 0, 1, 2, 3$ of $\pi/2$ so that ψ' lies within the appropriate quadrant $\left[-\frac{\pi}{4} + k\frac{\pi}{2}, \frac{\pi}{4} + k\frac{\pi}{2}\right]$. This adjustment operation is especially important when streamlines cross jump lines where ψ abruptly changes by $\pm\pi/2$, but the streamline has to continue in the same overall direction.

The ODE[2] (3.11) requires initial conditions in the form of

$$\vec{x}(t_0) = \vec{x}_0, \tag{3.12}$$

where t_0 is the initial integration time and $\vec{x}_0$ the location of the start of the streamline. In this case, the start of the streamline will either be an irregular node or a corner from which a non-boundary separatrix emerges. The ODE (3.11) can then be integrated in a traditional numerical manner as will be described below in Sec. 3.2.3.

[2] ODE: ordinary differential equation.

3.2 Implementation

From the discussion above, the decomposition of a domain into quadrilateral subdomains has four stages:

1. Computation of a guiding field. This includes the generation of a triangular mesh and solution of the Laplace problem (3.4).
2. Finding critical points of the guiding field and their valences, including the valences of corners at boundary vertices.
3. Integration of streamlines, from critical points and corners, to generate the interior boundaries of the quadrilateral decomposition.
4. Cutting of the domain into quadrilaterals using the separatrices as subdomain boundaries, then further subdivision into elements, as necessary.

This section presents details of the implementation of the process and walks through the four stages needed to generate a quadrilateral mesh. The procedure is illustrated using a geometry commonly used in the cross field literature [37, 84], that of a half disc as shown in Fig. 3.5.

3.2.1 Solution of the field equations

To solve the Laplace problem (3.4) the spectral/*hp* element formulation of Ch. 2 is used. A triangular finite element mesh is first generated which is then made high-order and curved by projecting interior nodes onto the curved boundaries [70]. This procedure is carried out in *NekMesh*, which also has the capability to optimise the high-order mesh should some elements be of low quality or simply invalid [80]. Such a mesh is shown in Fig. 3.5 for the reference geometry.

For the purposes of generating the quadrilateral decomposition, as coarse a triangular mesh as possible is generated and then high-order polynomials are used to approximate the solution within the elements to get the desired accuracy. First, it is desirable to take advantage of the spectral accuracy of the spectral/*hp* element Laplace solver, where a fine mesh is not necessary. Second, as will be discussed in Sec. 3.2.2 and Sec. 3.2.3, a small number of elements is desirable to simplify and speed up the critical point detection and streamline tracing. For the half circle geometry and triangle mesh shown in Fig. 3.5, it was sufficient to use third order polynomials (fourth order convergence with element size).

Once the curvilinear triangular mesh is generated, the Laplace problem (3.4) is solved. A special implementation of the Laplace solver has been implemented

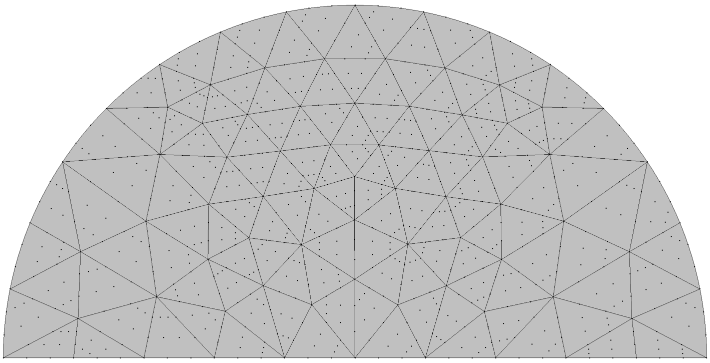


Figure 3.5: High-order triangular mesh on the half disc.

in *Nektar++* for the BVP (3.4), where the BC are automatically computed using Eq. (3.3).

Choice of a discretisation

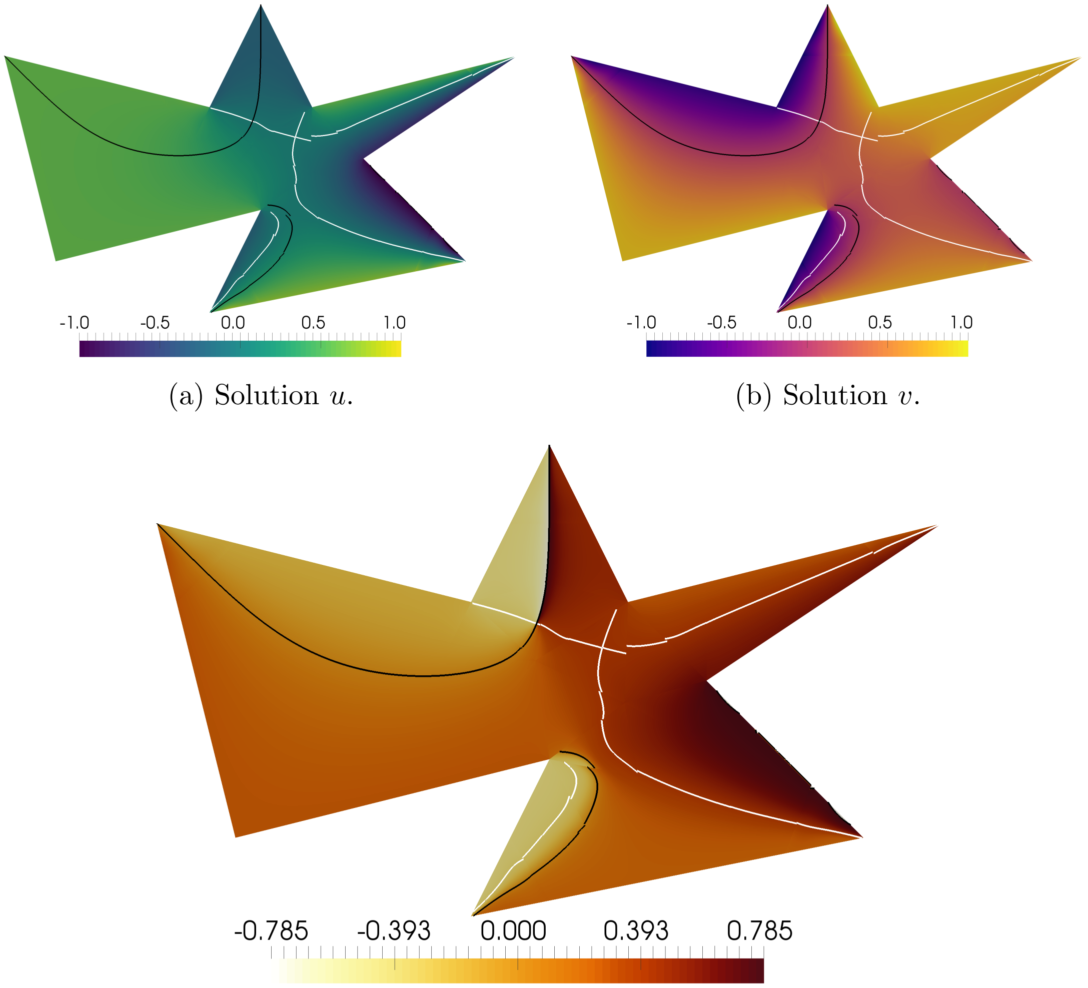
Nektar++ supports both CG and DG discretisations. The CG approximation is used when the BC are continuous along all boundary curves. As can be seen in the discussion in Sec. 3.1, BC are continuous if the boundaries satisfy one of two conditions:

- The curve at each point is C^1-continuous, i.e. it is a smooth curve; or
- The boundary curve is only C^0-continuous, i.e. a corner, and the angle is a multiple of $\pi/2$.

In this work, CG is employed whenever possible and DG only when the geometry requires discontinuous BC. For example, the reference geometry of Fig. 3.5 contains only smooth curves and $\pi/2$ corners. Therefore a CG formulation is used. The solution for $\vec{v}$ on the half disc geometry is shown in Fig. 3.4. As expected, BC and the solution are smooth, so the solution benefits from rapid convergence of the spectral/*hp* element methods.

If neither smoothness condition is satisfied at a boundary corner point, the BC (3.3) has a jump discontinuity there. In that case, a DG formulation is used, which can account for discontinuous BC. Unlike the CG approximation used in traditional methods, no *ad hoc* smoothing of a corner's BC is required, leading to a discretisation-consistent solution. It is important to note that a DG formulation may also be used at all times. The use of a DG formulation would especially be unaffected by a loss of C^1-continuity at boundaries with too coarse a discretisation. Indeed, smoothness of the boundary discretisation is only valid to a certain tolerance. As the resolution decreases, so does the smoothness to the point where BC become discontinuous. For this reason, the DG formulation is a more robust choice. The discussion and use of the CG formulation in this work are included for completeness.

As an example of a geometry that does not satisfy the smoothness conditions, a polygon geometry is used, shown in Fig. 3.6, which shows the solution and ψ fields. The triangular mesh used to obtain this solution is shown in Fig. 3.7. It is quite coarse, yet a solution is obtained with enough resolution, through higher polynomial order, to properly carry out the rest of the procedure.



(a) Solution u.

(b) Solution v.

(c) Computed ψ.

Figure 3.6: Guiding field on a polygon with arbitrary corners using a DG discretisation: isocontours of $u = 0$ and $v = 0$ shown in white and black respectively.

Using a DG formulation to handle corners with angles that are not multiples

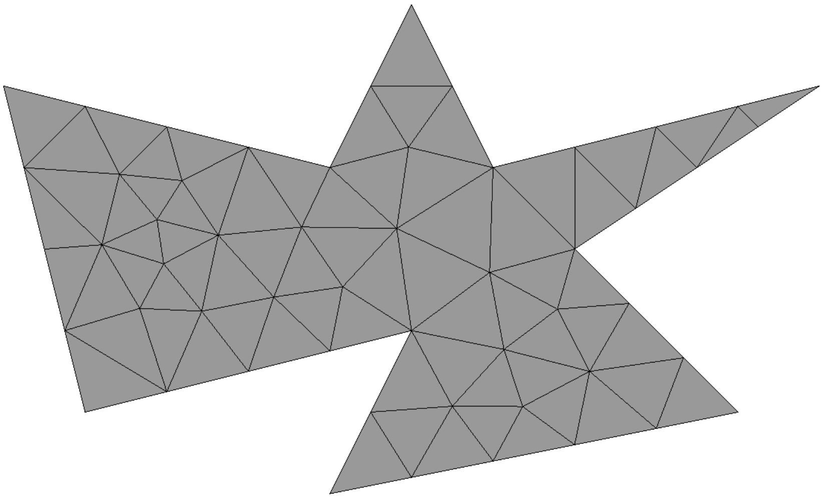
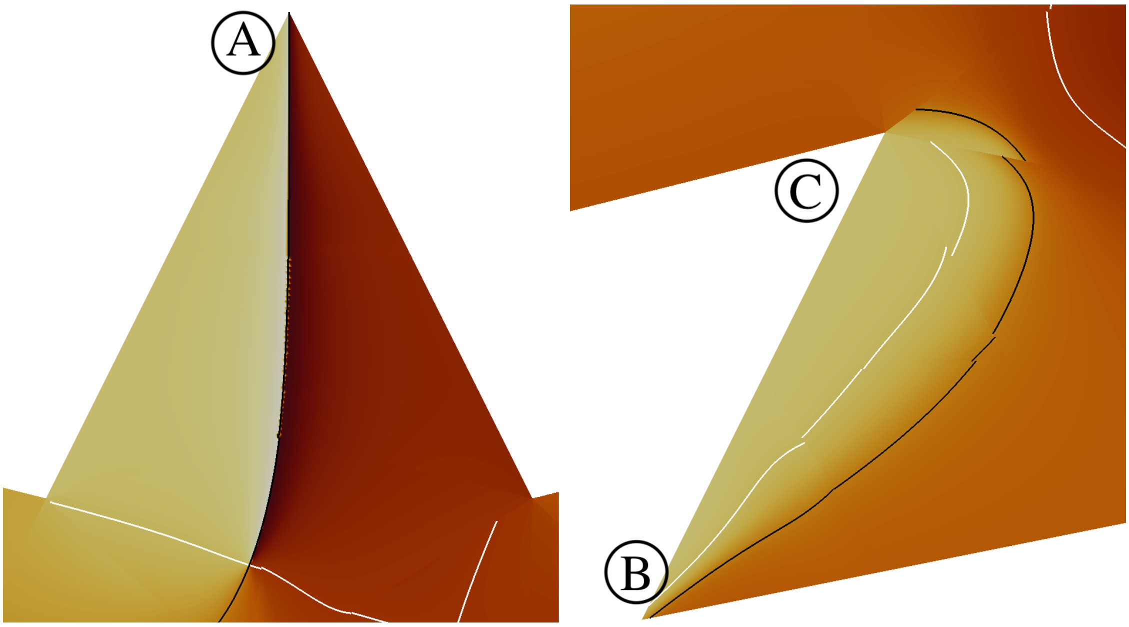


Figure 3.7: Triangular mesh on the polygon with arbitrary corners.

of $\pi/2$ has two consequences. First, jump lines can originate from corners. This fact is important, for it shows that one cannot *a priori* and *ad hoc* determine the valence of a boundary corner solely from its angle. Fig. 3.8a shows how the jump line originating from the top corner Ⓐ in the geometry ends in the sole critical point of the domain. Although curvature produces critical points, see Sec. 3.1.2, this straight-sided example shows that a corner whose angle is not a multiple of $\pi/2$ can also generate interior critical points, again, see Fig. 3.8a.



(a) Uppermost corner.

(b) Lowermost corner.

Figure 3.8: Guiding field ψ on the polygon: close-up views.

Second, observe that discontinuous BC are naturally enforced in the DG formulation. No *ad hoc* smoothing of the boundary is required, as shown in Fig. 3.8b. At a convex corner (see lower corner, Ⓑ), both boundary edges are shared by a single element. In this case, the BC are enforced through fluxes between the boundary edge elements and the interior volume element so that the solution remains continuous inside this interior volume element. This is clearly seen in Fig. 3.8b where the solution is smoothed out at the corner, in a formulation consistent manner. At a concave corner (see upper corner, Ⓒ), however, the boundary edges are not shared by a single element and the Dirichlet BC can be enforced exactly. This naturally leads to a discontinuity in the solution between elements in the vicinity of the corner.

When a DG method is used to solve the Laplace problems for the guiding fields, the fields are discontinuous between elements. This is especially true in domains where discontinuous BC are present, as is clear in Fig. 3.8b. Enough resolution (either through h- or p-refinement) needs to be provided so that the jumps at the element interfaces are small. Too low resolution could result in zero isocontours located within solution jumps between elements. The current implementation would not be able to detect such isocontours, nor could it consequently detect critical points. See Sec. 3.2.2 for more details.

Fortunately, critical points should naturally be located in regions of smooth fields where jumps between elements are small. Discontinuities between elements are large only near boundaries where discontinuous BC appear. The field on the inside of the domain is mostly smooth, due to the properties of the Laplace problem, resulting in very small jumps between elements. If strong curvature is present, critical points can appear near curved boundaries where BC are continuous and the field is smooth. If no (or little curvature) is present, critical points are controlled by discontinuous BC and are seen far away from the boundaries, on the inside of the domain, again where the field is smooth. In all cases, critical points are not expected to naturally occur in areas where large jumps between elements exist due to nearby discontinuous BC.

Subsequently, the integration of streamlines is unaffected by the discontinuous nature of the discretisation. By using a multi step high-order integrator, as explained later in Sec. 3.2.3, streamline direction is not strongly affected by abrupt changes of direction between elements. Streamlines remain smooth even when gradients in the guiding field are large, e.g. when crossing a large jump between elements.

3.2.2 Detection of critical points

Once the solution to Eqs. (3.4) is computed, zeros in the field $\vec{v}$ are found. Unlike the traditional cross field approach, crosses are in fact never generated in this method. As seen in Sec. 3.1 the analysis can be performed on the $\vec{v}$ field, and ψ is computed only when necessary, i.e. for computing the valence of vertices and tracing streamlines via Eqs. (3.10) and (3.11). All other operations can be accomplished by operating on $\vec{v}$.

Critical points inside the domain are located at $\vec{v} = \vec{0}$. To find those, each element that may contain a $\vec{v} = \vec{0}$ point is first flagged. These correspond to elements crossed by both a $u = 0$ and a $v = 0$ isocontours. To contain a $u = 0$ isocontour, there must be at least one quadrature point with value $u \leq 0$ on one side of the isocontour and one quadrature point with value $u \geq 0$ on the other side of the isocontour. The argument is the same for a $v = 0$ isocontour. If these conditions are fulfilled, the element is flagged. Then, in each flagged element, Newton's method is found to approach the location of $\vec{v} = \vec{0}$. To do so efficiently, the critical point search is performed in parametric space. If a critical point is found outside the reference element, this element is dismissed as it is assumed that it will be found through parametric search in a neighbour element. In Fig. 3.4, the two symmetrically located critical points can be clearly seen where the black and white contours cross, consistent with the irregular nodes found by others, e.g. [37], for this geometry.

Calculation of valences

After completing the search for all critical points, a valence must be computed for each. From Sec. 3.1.2 and Eq. (3.8), the integral I_c can be evaluated solely by looking for the presence of jump lines. In a counter-clockwise manner, a positive jump (i.e. from $-\frac{\pi}{4}$ to $\frac{\pi}{4}$) leads to a negative integral $I_c = -1$, and vice versa. The sign of the jump can itself be determined solely by values of $\vec{v}$, without the need to compute ψ or the cross field.

A jump must satisfy two conditions: u must be negative and v must change sign. A positive increase of v indicates a positive jump, and vice versa. To detect jumps, and therefore compute the valence of a critical point, a sequence of uniformly distributed quadrature points located on a circle of small radius c, centred on the critical point, is used to step counter-clockwise. At the present time, this parameter c is chosen empirically to be 10^{-6}. This value has little importance on the accuracy of the evaluation of the integral. Importantly, the circle formed by c around the critical point should not encircle any other critical point. This would lead to a wrong valence. Following this logic, the critical points seen in Fig. 3.4 are both determined to have a valence of three.

The valences of all boundary vertices are computed in a similar fashion, this time from expression (3.10). The angle at the corner, determined from θ_0 and θ_f, is computed from the geometry. To compute the integral $I\left(\theta_0, \theta_f\right)$, $\Delta\psi = \psi_f - \psi_0$

is found from the values of ψ at the boundaries using the BC. If present, a jump contribution is added to $\Delta\psi$ in the same way as for the computation of critical point valences. Because both corners in Fig. 3.4 have an angle of $\frac{\pi}{2}$, a valence of one can be trivially determined.

It is important to note that the boundary vertex operation does not require an *a priori* determination of the corner valence based on angle only. The corner valences can be determined through Eq. (3.10) which does not only depend on the *known* BC but also on the existence of jump lines in the *computed ψ field* in the neighbourhood of the vertex, like that seen in Fig. 3.8.

The boundary vertex valence gives the number of quads located at the associated corner. Due to the definition, this valence could be zero. Indeed, $\Delta\theta$ and $\Delta\psi$ could cancel out if the corner is sufficiently sharp. Physically, this indicates the presence of a degenerate quadrilateral block (i.e. a triangular block) where all streamlines would converge towards the degenerate corner. This topology requires *ad hoc* manipulation where the streamlines are used later to construct the quadrilateral subdivision of the domain. This manipulation will be explained in the next section.

3.2.3 Integration of streamlines

Streamlines are traced throughout the domain after the valences of all irregular nodes and vertices are determined. The first step is to find the initial direction of each streamline. Because $\vec{v} = \vec{0}$ at critical points, ψ is undefined. Therefore, streamline angles are determined from $\vec{v}$ at a small distance c of the critical point. This distance is currently set empirically based on the size of the elements.

The initial direction is refined iteratively from an initial guess using Alg. 1, which is inspired by bisection. For the first streamline, an arbitrary initial guess is taken for the direction. For subsequent streamlines, an initial guess is taken at angles multiples of $\frac{2\pi}{\mathcal{V}}$. The size of the tolerance, ϵ is not critical; in this work, $\epsilon = 10^{-9}$.

The angle search in Alg. 1 can also be applied to boundary vertices, including those where the BC might be discontinuous and ψ is ambiguous. An initial guess for *all* streamlines may be obtained at angles multiples of $\frac{\Delta\theta}{\mathcal{V}}$.

After an initial direction is obtained, streamlines are *synchronously* advanced throughout the domain. This part of the procedure is the most computationally expensive. Each new streamline point requires the search for the element that contains it. Next, the inverse map $\vec{\xi} = \mathcal{X}_e^{-1}(\vec{x})$ is used to transform to parametric coordinates. Finally the high-order interpolation of $\vec{v}$ is computed via Eq. (2.4).

A 4^{th} order multi step Adams-Bashforth integrator is used to advance the streamlines. The current implementation uses fewer points at the beginning of the integ-

```
Data: Direction $\alpha_0$ (initial guess), irregular node location $\vec{p}_0$
Result: Direction $\alpha_f$ (converged value)
Initialise $\alpha$ from $\alpha_0$ ;
while $|\Delta\alpha| > \epsilon$ do
    Compute point $\vec{p}_1$ at distance $c$ and direction $\alpha$ of $\vec{p}_0$ ;
    Interpolate $\vec{v}$ and compute $\psi$ at $\vec{p}_1$ ;
    Find $\psi' = \psi + k\frac{\pi}{2}$ where $k = 0, 1, 2, 3$ such that $\Delta\alpha = |\alpha - \psi'|$ is
      minimised ;
    Update $\alpha = \psi'$ ;
end
Return $\alpha_f = \alpha$ ;
```

Algorithm 1: Initial direction of a streamline at an irregular node.

ration when the point history of the streamline trace is limited. While advancing the streamlines, it is important to validate the direction α obtained at the latest point. When a streamline crosses a jump line, ψ abruptly rotates $\pm\pi/2$ and α must be adjusted accordingly.

Manipulation of streamlines

As streamlines are advanced throughout the domain, they may meet. To anticipate two streamlines meeting, the front points of each are compared at each step. If the distance between the front points of two streamlines is less than the step size d and they are advancing in opposite directions, the streamlines are assumed to meet. When the absolute value of the difference in the latest α values equals π when rounded to the nearest $\pi/2$, the streamlines are considered to be advancing in opposite directions. This process is summarised in Alg. 2.

When two streamlines meet, they are advanced to the starting point of the other, keeping the same number of points in each, before merging. See Alg. 3 for more details. A merged streamline is created by weight-averaging a pair of streamline points. A simple linear weight function could be used, but that will change the angles computed with Alg. 1 at the starting points, both irregular nodes and boundary vertices. This not desirable as the PDE problem has already optimised these angles to be as equally distributed as possible. Instead trigonometric weight functions of the form $W_0(x) = \cos^2(x)$ and $W_1(x) = \sin^2(x)$ are used, where $x \in [0, 1]$ is the parametrisation along the streamline. Because $\frac{dW_0}{dx}$ and $\frac{dW_1}{dx}$ are zero at $x = 0$ and $x = 1$, angles are preserved at the initial and final points of the merged streamline.

```
Data: Starting point $\mathbf{x}_0$ and initial direction $\alpha$ for each streamline, step
      size $d$
Result: List of points $\vec{\mathbf{x}}$ for each final, merged streamline
repeat
    forall incomplete streamline $i$ do
        Advance streamline $i$ by one step ;
        Push new point $\mathbf{x}$ to the front of the list, becoming $\mathbf{x}_0$ ;
        if $\mathbf{x}_0$ outside of domain then
            Mark streamline $i$ completed ;
            Skip to the next streamline $i$ ;
        end
        Update $\alpha_i$ ;
        forall incomplete streamline $j \neq i$ do
            if $|\mathbf{x}_{i,0} - \mathbf{x}_{j,0}| < d$ and $\mod (|\alpha_i - \alpha_j|, 2\pi) = \pi$ then
                Merge streamlines $i$ and $j$ into streamline $k$ ;
                Delete streamlines $i$ and $j$ ;
                Mark streamline $k$ completed ;
                Skip to the next streamline $i$ ;
            end
        end
    end
until all streamlines are completed;
Return $\vec{\mathbf{x}}$ for each streamline ;
```

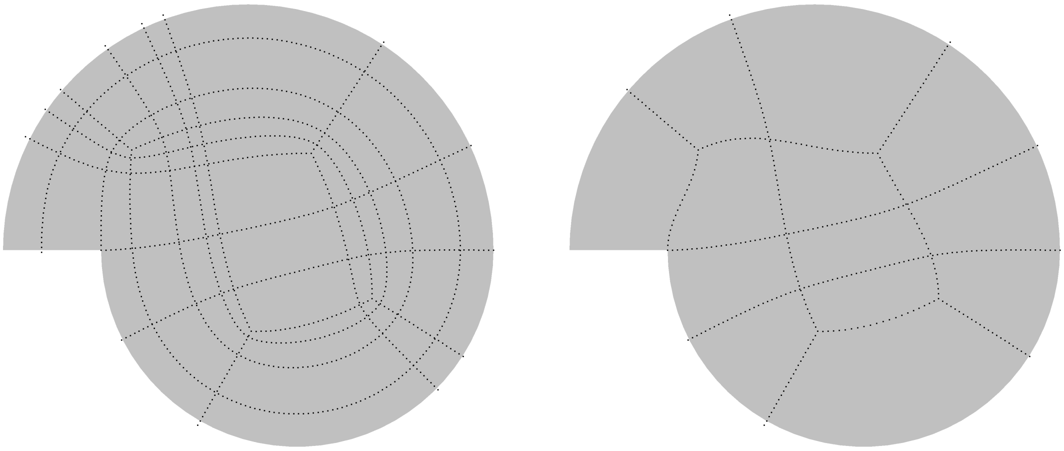
Algorithm 2: Advancing of streamlines.

Aggressive merging

What has been described above constitutes expected or *normal* merging. But the user may also want to reduce the number of separatices and thereby simplify a valid block decomposition. One approach is to perform *aggressive* merging of the separatrices. For aggressive merging, a larger distance threshold is carefully chosen to make streamlines that would normally just miss each other merge anyway. Trigonometric weight functions are especially useful when performing aggressive merging of two reasonably distant streamlines. An example of aggressive merging with trigonometric weight functions is presented in Fig. 3.9 where Fig. 3.9a corresponds to normal merging and Fig. 3.9b to aggressive merging. In this case, a distance threshold of 5 times the step size was used for aggressive merging. This example shows that the separatrix graph is greatly simplified and that the curving of some streamlines helps preserve the original direction at each end. Note, importantly, that irregular nodes obtained from the critical points are not modified *a posteriori*; only separatrices are.

```
Data: List of points $\vec{\mathbf{x}}$ and latest direction $\alpha$ for each of streamlines $i$ and $j$
Result: List of points $\vec{\mathbf{x}}$ for the merged streamline $k$
Advance streamline $i$ by length $(\vec{\mathbf{x}}_j) - 1$ and streamline $j$ by length $(\vec{\mathbf{x}}_i) - 1$ ;
$N = \text{length}(\vec{\mathbf{x}}_i)$ ;
Invert list $\vec{\mathbf{x}}_j$ ;
forall $t = 0, 1, \ldots, N-1$ do
    $\mathbf{x}_{k,t} = \cos^2\left(\frac{t}{N-1}\right)\mathbf{x}_{i,t} + \sin^2\left(\frac{t}{N-1}\right)\mathbf{x}_{j,t}$ ;
end
Return $\vec{\mathbf{x}}_k$ ;
```

Algorithm 3: Merging of two streamlines.



(a) Normal merging.

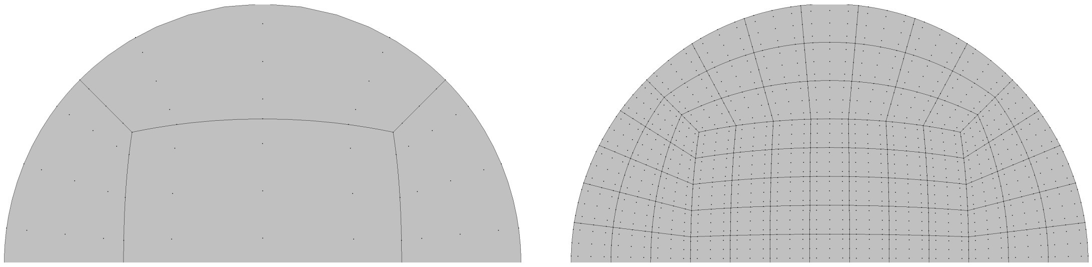
(b) Aggressive merging.

Figure 3.9: Merging of streamlines.

3.2.4 Quadrilateral decomposition

NekMesh relies on the *Open Cascade* platform [58] as its CAD engine for mesh generation and for the projection to high-order boundary representations. *Open Cascade* also includes tools for CAD manipulation. These capabilities are used, in this work, for spline and wire creation and for face splitting.

The separatrices computed from the guiding field are transformed into *Open Cascade edges* represented by interpolating splines. These *edges* can be joined into a set of *wires*, with each *wire* consisting of topologically connected *edges*. The original CAD file is loaded again, consisting of a single two dimensional *face*, representing the domain Ω. This *face* is then iteratively split by each wire and a set of quadrilateral *faces* is obtained. These faces are topologically connected, meaning that the future mesh will be conforming. At this point, each face can be trivially meshed with a single quadrilateral and projected to high-order. Fig. 3.10a illustrates the quadrilateral decomposition and coarse quadrilateral mesh obtained for the half circle after computing the location of irregular nodes and tracing the separatrices.



(a) Coarse mesh.

(b) Split mesh.

Figure 3.10: Quadrilateral meshes on the half disc.

For this coarse mesh, the worst scaled Jacobian is found to be $J_{s,\min} = 0.490$. In this chapter, values for the worst scaled Jacobian are typically only reported for the coarse mesh. Indeed, as the mesh is split, the quality of elements improves and the worst scaled Jacobian increases. In this case for example, the worst scaled Jacobian goes up to $J_{s,\min} = 0.965$ for the split mesh shown in Fig. 3.10b.

The midpoint division approach [39] is used to split a triangular block that forms when a corner valence is zero. An artificial 3-valence irregular node is inserted which is connected to each of the three sides of the triangle. One separatrix can then be physically integrated away from the degenerate corner and throughout the domain. The other two branches of the 3-valence irregular node are defined as straight lines at $\pm\frac{2\pi}{3}$ from the physical separatrix. Fig. 3.11 shows what the midpoint division of a degenerate quadrilateral might look like.

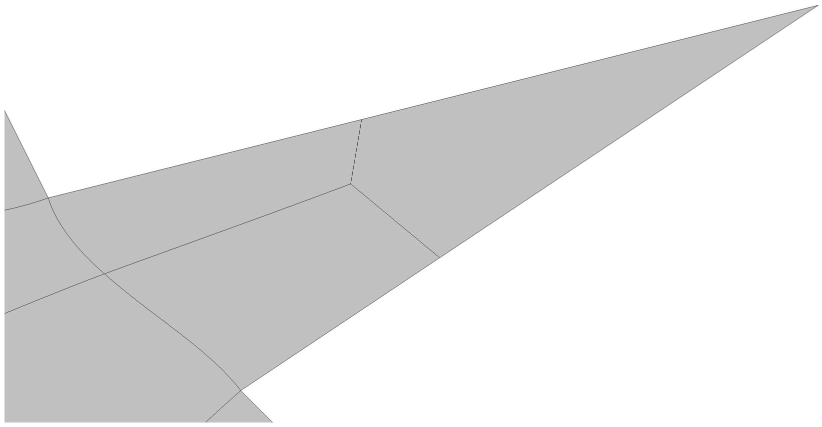


Figure 3.11: Midpoint division of a degenerate quadrilateral: close-up view of Fig. 3.17.

3.2.5 Bidirectional isoparametric refinement

If desired, an isoparametric splitting approach [54] can be used to further subdivide the quadrilateral subdomains into smaller elements. This technique, initially developed for boundary layer mesh division, guarantees that subdivided elements have the same quality as the macro element from which they are created. In the current context, it also ensures that angles are preserved.

Since elements are valid thanks to the nature of cross fields, there exists a bijective mapping χ between a reference element Ω_{st} and the physical space element Ω. The mapping is used to introduce subdivisions, according to a user-defined criterion, of the reference element along the height to generate layers in the physical space, as shown in Fig. 3.12. This way very thin elements can be generated that are themselves valid if the mapping satisfies certain restrictions [53, 54].

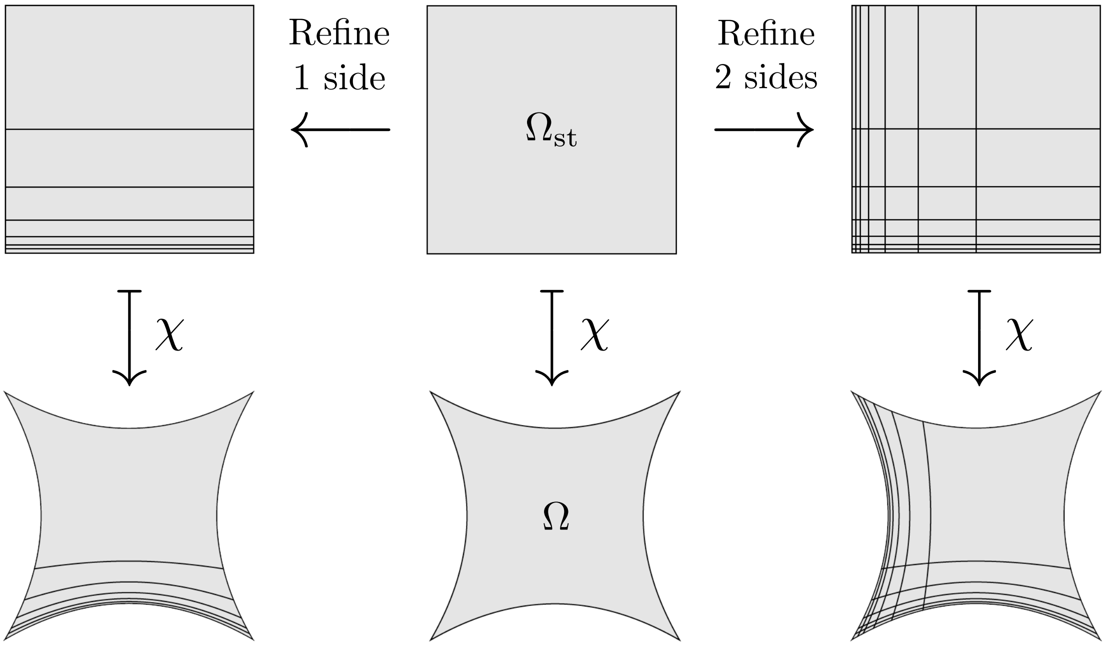


Figure 3.12: Adaptation of the isoparametric splitting for *two* directions.

The original isoparametric refinement technique [54] splits a valid prismatic macro-element into a stack of high-order prismatic elements by using the polynomial mapping that defines the curvature of the element. Mathematically, this mapping $\chi : \Omega_{\mathrm{st}} \to \Omega$ is defined between a reference element Ω_{st} and a given element Ω. The key observation in the isoparametric splitting technique is that, to generate the stack of elements, the standard element Ω_{st} may be split into reference sub-elements, and then χ applied to these to produce sub-elements in Cartesian space. This process is depicted visually on the left-hand side of Fig. 3.12 for a simple quadrilateral element.

To apply this technique for the present problem of quadrilateral block refinement, an adapted version of this approach is required wherein the reference element is split in not one but *two* directions. From the perspective of the mathematical justification for the validity of the method, this aspect actually makes very little difference. As noted in reference [54], the splitting of the reference element in the original isoparametric technique can be viewed as an affine mapping $f : \Omega_{\mathrm{st}} \to \widetilde{\Omega}_{\mathrm{st}}$, where $\widetilde{\Omega}_{\mathrm{st}}$ is a sub-element of Ω_{st}. The curvature mapping of a sub-element of the Cartesian element $\widetilde{\Omega}$ can then be viewed as the composition $f \circ \chi : \Omega_{\mathrm{st}} \to \widetilde{\Omega}$. Then, as long as f is defined such that its Jacobian determinant $J_f(\xi) > 0$ for all $\xi \in \Omega_{\mathrm{st}}$, then this new mapping is valid so long as χ is also valid.

To adapt this technique for bidirectional splitting, a slightly different refinement strategy is therefore required, as depicted on the right-hand side of Fig. 3.12, where the standard element is split in each direction. A small extension to the method has been applied which computes the orientation of the quadrilaterals and alters the distribution of the splitting points in the reference element accordingly, but the core of the method remains mostly the same.

Each row of quads can now be individually split using the isoparametric approach. The fine mesh in Fig. 3.10b shows what the coarse mesh in Fig. 3.10a looks like after a conformal splitting of each row of quads.

3.3 Application

Five geometries are presented here taken from, or inspired by, the literature, for which meshes are generated. The meshes shown in Fig. 3.10 are typical of those already generated by traditional approaches [37, 84, 83]. The additional examples are used to illustrate the application of the high-resolution approach on multiply connected geometry, one for which traditional approaches have been shown to have difficulty, a polygon with both acute and obtuse angles, a NACA[3] 0012 aerofoil, and a gear.

3.3.1 Geometry I: multiply connected rectangle

Geometry I shown in Fig. 3.13 is a rectangle with two quarter circle holes. This geometry has also been meshed, for example, in reference [37]. Some traditional implementations have produced spurious asymmetric irregular nodes, see Fig. 3.13 [32], for example. The high-order guiding field approach, however, produces the minimum

[3] NACA: National Advisory Committee for Aeronautics.

number of irregular nodes (two 3-valence irregular nodes) while preserving the symmetry of the geometry without the need to coalesce spurious irregular nodes that can result from traditional detection algorithms. Fig. 3.14 shows the coarse quadrilateral mesh obtained on this geometry using the guiding field approach. The worst scaled Jacobian for this coarse mesh is $J_{s,\min} = 0.786$.

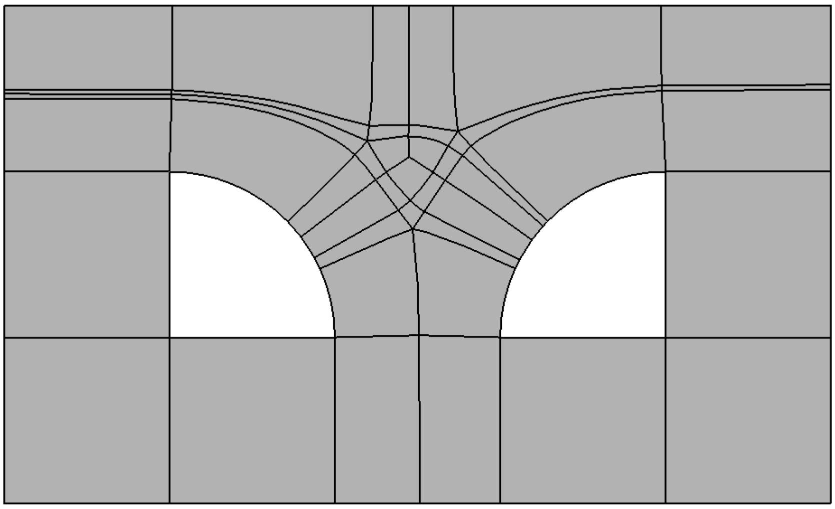


Figure 3.13: Block decomposition on Geometry I with spurious asymmetric irregular nodes, generated with a traditional implementation [32].

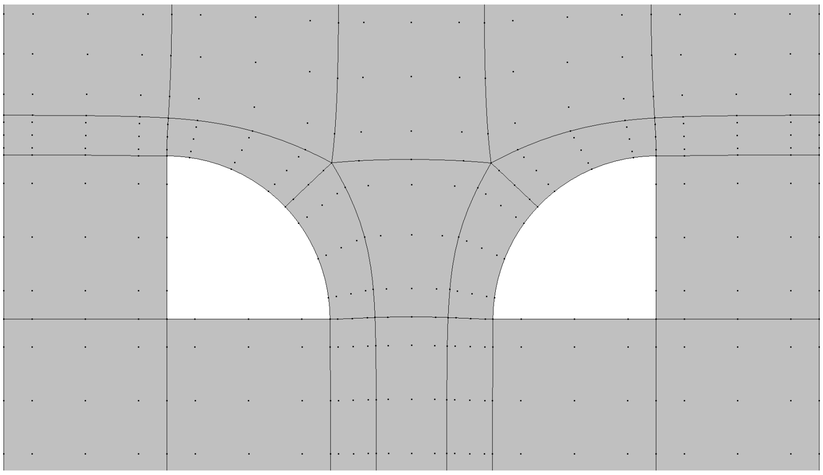


Figure 3.14: Coarse quadrilateral mesh on Geometry I using the high-resolution guiding field approach.

3.3.2 Geometry II: nautilus

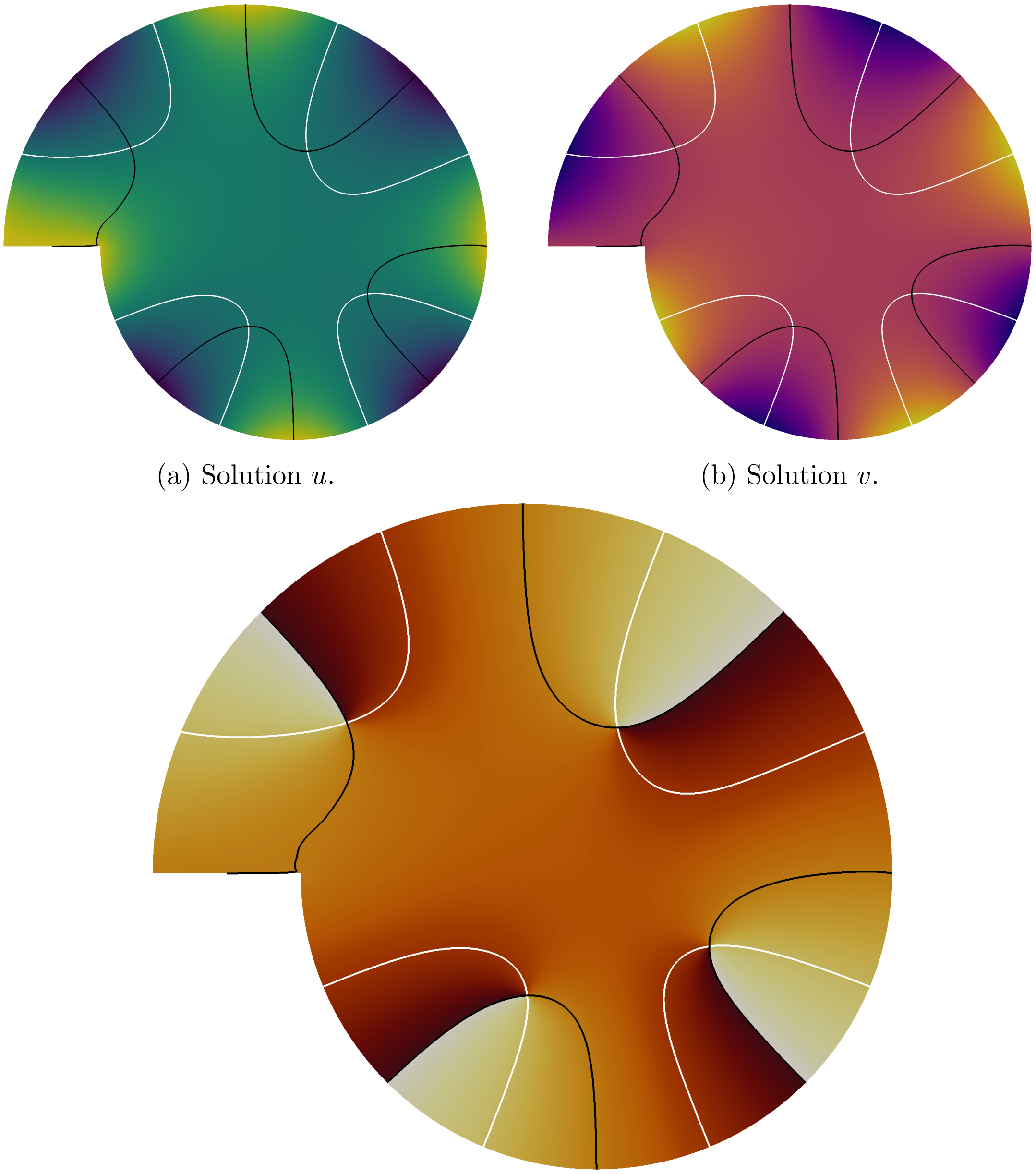
Geometry II is the nautilus. The domain and guiding fields shown in Fig. 3.15 give the decomposition shown in Fig. 3.16 when aggressive merging is applied. This domain is one that traditional methods generate spurious spiral separatices that end in a limit cycle, therefore creating an invalid block decomposition [84, 83]. To date, no approach using cross fields has been presented that generates a valid separatrix graph. The use of high-order integration of an accurate guiding field allows *NekMesh* to generate separatrices that do not form a spiral. In fact, the irregular nodes found by *NekMesh* on this geometry are identical to those one would obtain for a simple disc. Fig. 3.16 shows the coarse quadrilateral mesh obtained on this nautilus geometry. The worst scaled Jacobian for this coarse mesh is $J_{s,\min} = 0.671$. This decomposition was generated using aggressive merging with a distance threshold of value 5 times the step size. If normal merging was to be used, the block decomposition would have looked more complicated, as shown in Fig. 3.9a, reducing the flexibility for users to split the mesh to their liking.

The nautilus has a similar (mirrored) guiding field to that obtained on the half disc in Fig. 3.4. The nautilus itself consists of two half discs of different size attached by their chord. The irregular node pattern (four 3-valence nodes) is therefore expected. One major difference is found with the irregular nodes obtained with previous techniques, e.g. references [84, 83]: the upper left irregular node is located further outside than in previous works. This seems to lead to streamlines successfully escaping a limit cycle.

3.3.3 Geometry III: polygon

Geometry III consists of a polygon whose corners include acute and obtuse angles. This example is used to demonstrate the use of the DG discretisation for the solution of the guiding field. The rest of the meshing procedure remains the same and is in fact unaffected by the type of discretisation. Fig. 3.6 has already shown the solution $\vec{v}$ and computed ψ field for this geometry, with $u = 0$ and $v = 0$ isocontours shown in white and black, respectively. As mentioned earlier in Sec. 3.2.3, the only consideration when using the DG formulation is that the approximate solutions are accurate enough to not affect the streamline integration.

Geometry III features a sharp corner (far right, Fig. 3.17) whose valence is evaluated as zero. That creates a degenerate quadrilateral that must be split into three valid quads as explained in Sec. 3.2.3. Fig. 3.17 shows the coarse quadrilateral mesh obtained on the polygon geometry. It contains only two irregular nodes, one created *ad hoc* for the degenerate corner and the other detected as a critical point in the guiding field. The worst scaled Jacobian for this coarse mesh is $J_{s,\min} = 0.392$.



(a) Solution u.

(b) Solution v.

(c) Computed ψ.

Figure 3.15: Guiding field on Geometry II: isocontours of $u = 0$ and $v = 0$ shown in white and black respectively.

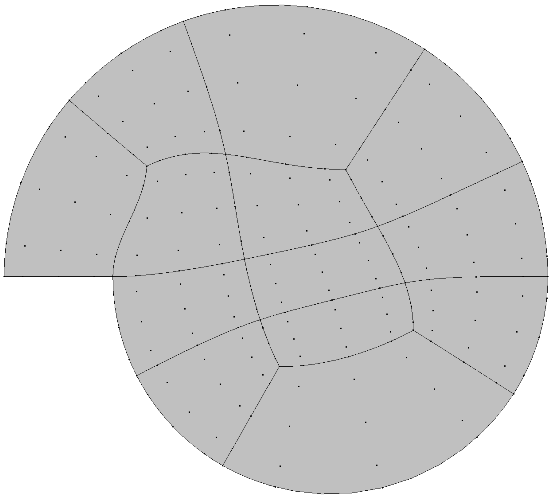


Figure 3.16: Coarse quadrilateral mesh on Geometry II.

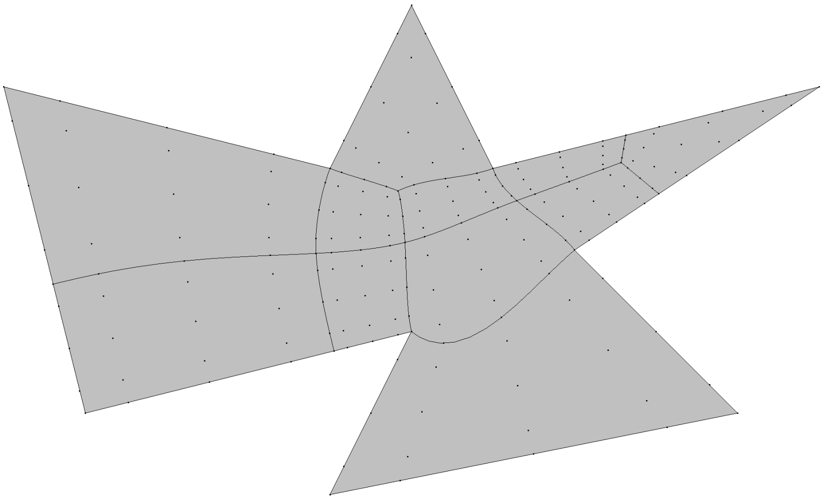


Figure 3.17: Coarse quadrilateral mesh on Geometry III.

3.3.4 Geometry IV: NACA 0012

Geometry IV is that of a NACA 0012 profile in a rectangular domain, more relevant for CFD[4] applications. Since the trailing edge angle is not a multiple of $\pi/2$, a DG discretisation was used to compute the guiding field. With this example, the flexibility of the isoparametric splitting module in *NekMesh* to define different distributions of elements is demonstrated. This allows the user to easily obtain elements of the required size, e.g. high aspect ratio elements in the boundary layer of the aerofoil. Fig. 3.18 shows the coarse and the split quadrilateral meshes obtained on this NACA 0012 geometry. The worst scaled Jacobian for the fine mesh is $J_{s,\min} = 0.084$, which is abnormally low, suggesting a potential issue in the mesh.

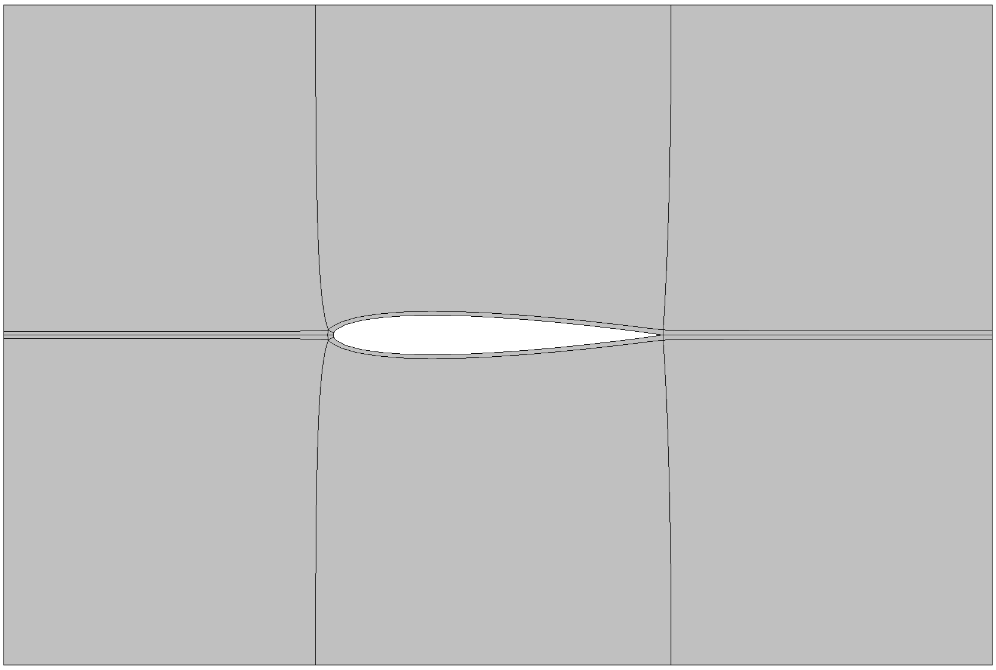
3.3.5 Geometry V: gear

Finally, Geometry V (Fig. 3.19) illustrates the preservation of symmetries and patterns. The geometry is that of a simple gear and the same irregular node pattern as seen in the cross field literature [83] is observed.

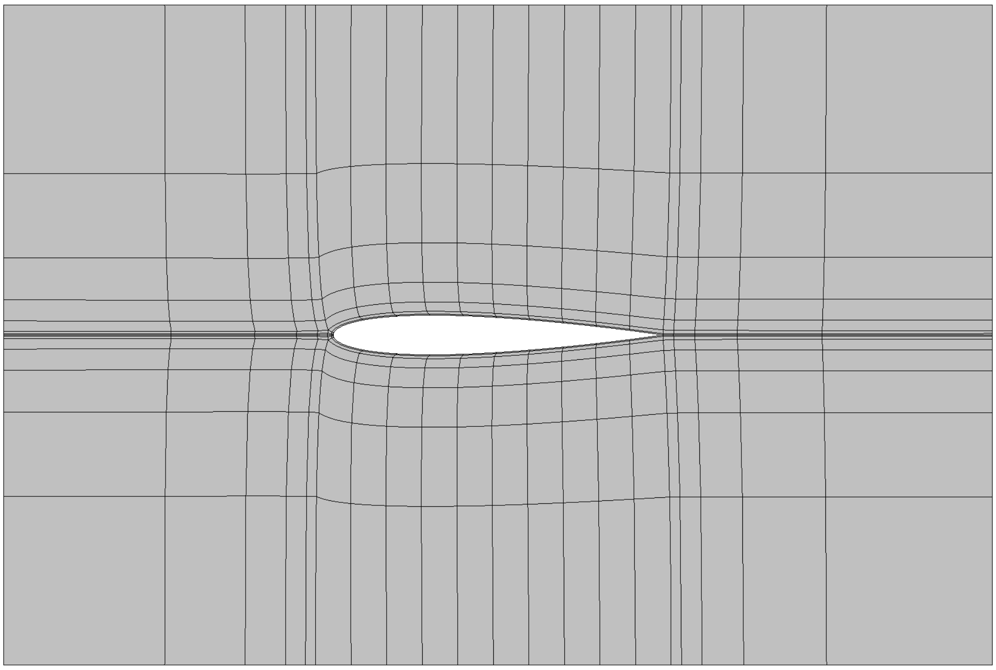
3.3.6 Fine meshes

The coarse meshes can be split further according to the user's preference using the isoparametric splitting described in Sec. 3.2.5. Examples of two split meshes for some of the quadrilateral decompositions obtained above are shown in Fig. 3.20.

[4] CFD: computational fluid dynamics.



(a) Coarse mesh.



(b) Split mesh.

Figure 3.18: Quadrilateral meshes on Geometry IV.

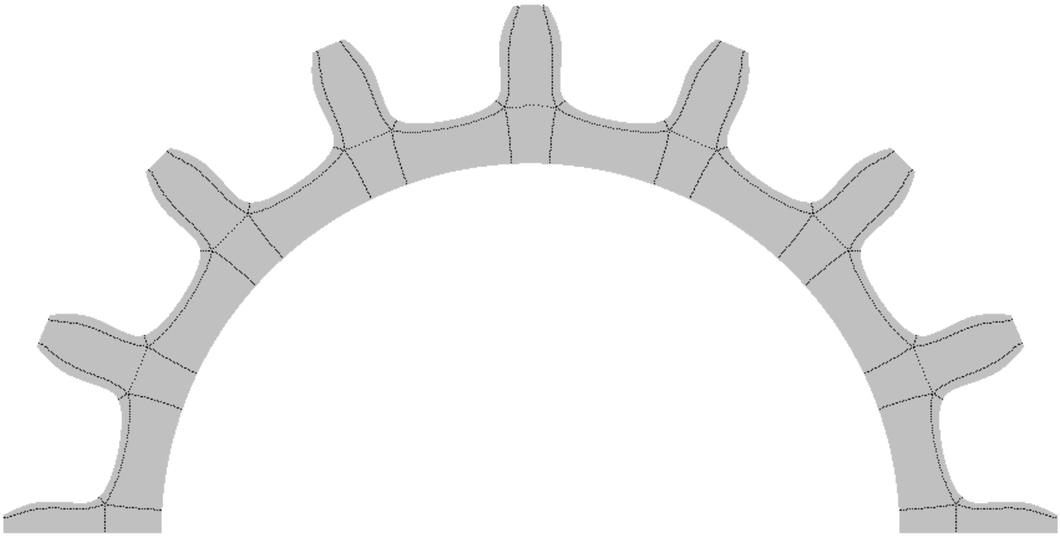
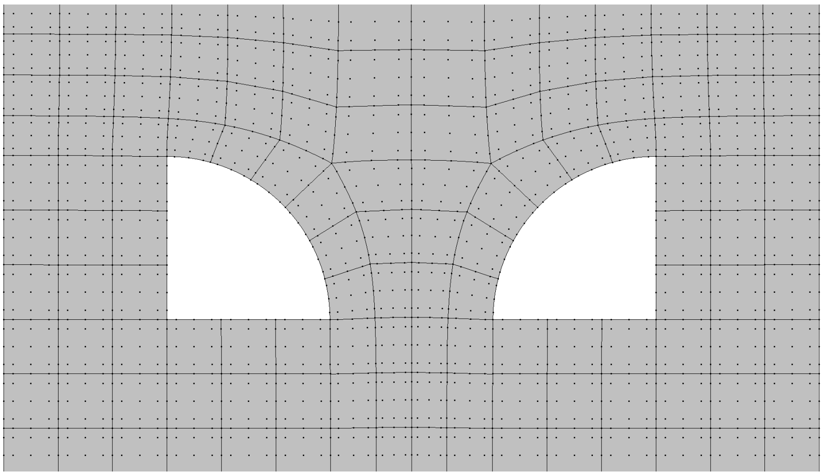
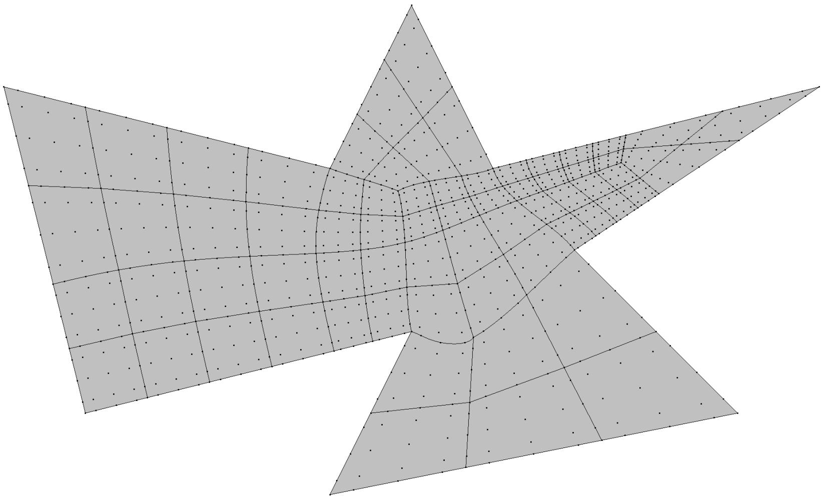


Figure 3.19: Block decomposition on Geometry V.



(a) Geometry I.



(b) Geometry III.

Figure 3.20: Split meshes on Geometries I and III.

Chapter 4

Three-dimensional hybrid mesh generation

The method presented in the previous chapter is well defined in two dimensions but does not translate well to three dimensions because of the lack of rigorous frame arithmetic. A semi-structured approach is instead pursued in this chapter where a structured mesh is generated in the near-field and an unstructured one in the far-field. This yields several advantages. A global hexahedral block decomposition is not sought, a task that remains a challenge for the community. Instead, a near-field partition is generated using the medial axis description of the geometry. The medial axis is a technique that allows geometries to be analysed such that smaller regions of simple shape can be created and a skeleton built [3]. This near-field partition yields a coarse boundary layer mesh that can be easily curved and refined in the wall-normal direction. The rest of the domain is then discretised using traditional unstructured mesh generation techniques. This semi-structured approach brings the benefits of both approaches together: high anisotropy is achieved near boundaries to better capture viscous effects; and no global block decomposition of the whole domain is needed.

The layout of this chapter goes as follows. The *CFI* CAD interface implemented in *NekMesh* is first described in Sec. 4.1. The medial axis is then described in Sec. 4.2, which is used for near-field partitioning and for the generation of the coarse linear mesh. A fine high-order mesh is then generated using a bottom-up approach, in Sec. 4.3, with some examples shown in Sec. 4.4. Finally Sec. 4.5 covers a recent extension of the method to generate structured meshes in the wake of streamlined bodies.

4.1 CAD interface

Processes for both linear and high-order meshing regularly interrogate the CAD geometry and thus a robust CAD interaction is required. *NekMesh* provides a lightweight wrapper that hides the complexity and size of the CAD interface from users and developers. In the examples presented here *CFI*, the CAD interface of *CADfix* [33], has been used but *NekMesh* also provides a CAD back-end to *Open Cascade* [58] as its CAD engine.

The use of *CADfix*, and its interface *CFI*, is motivated by the more stringent requirements on CAD quality for high-order meshing. CAD representations that may work very well within linear mesh generators, may not work for their high-order counterpart. For example, distortion levels in the surfaces, which might be perfectly acceptable for generating linear meshes, could induce poor quality or invalid elements in high-order meshes. Therefore access to high quality CAD and CAD repair tools for poor quality CAD, along with a robust CAD interface, is vital to the creation of robust tools for quality high-order meshing.

The flowchart of Fig. 4.1 depicts the integration of *CFI* into *NekMesh*. In a nutshell, a *CADfix* session produces a linear mesh that *NekMesh* will read via *CFI* and process through its own high-order routines. More details of the method will be given in the following sections.

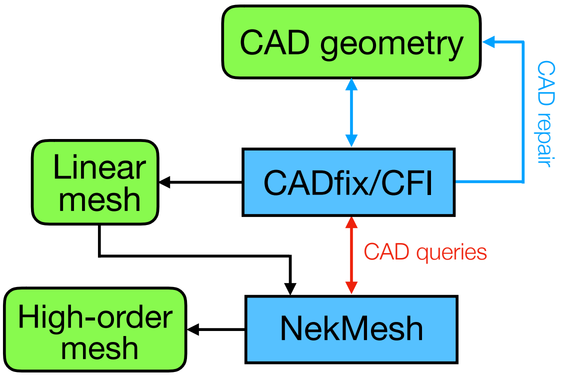


Figure 4.1: Flowchart of the proposed approach.

4.2 Linear mesh

In order to produce a high-order mesh, a linear mesh first needs to be generated. *CADfix* is a commercially available tool with functionality covering the import, preparation and interrogation of geometry but the 3D medial object based partitioning and linear mesh generation uses results from active research projects [11] and are not yet commercially available. However, in order to provide appropriate geometry and prismatic meshes for upgrade to high-order, *CADfix* has additional functionality which has been designed for this framework and is under active development. There are several automatic and semi-automatic tools which are included in the pipeline for generating linear meshes. First, the geometry is prepared to repair any CAD defects and to define a valid domain. Second, the domain is automatically subdivided, using the medial axis, to create partitions for meshing. Finally, edges and faces of the partitions are meshed using a coarse set of divisions, designed to be balanced, well aligned and to allow periodicity at the boundary. Each part of this process has been designed to be suitable for *a posteriori* high-order mesh generation. The methodologies presented in this section are available in *CADfix* and have not been implemented as part of this project. They are presented here for completeness.

To illustrate the various steps of the procedure for constructing a high-order mesh, a simple geometrical domain is used that consists of an unswept wing of rectangular planform composed of NACA 0012 aerofoil sections and a round tip, essentially a wing tip, enclosed in a rectangular box. This geometry is also of aerodynamic interest as a case study of vortex roll-up proposed and experimentally measured by Chow et al. [18] which has been used in CFD validation studies, see for instance [42].

4.2.1 Preparation

Not all CAD models are suitable for CFD analysis. CAD geometry often lacks far-field boundary definitions, it may have defects such as sliver surfaces or small gaps and the geometry may not be watertight. For meshing purposes a *CFD-ready CAD geometry* is required: the fluid domain must be a watertight CAD solid. *CADfix* can import the geometry from a wide range of design sources and provide automatic, manual and diagnostic driven tools for repairing poor quality CAD geometry, constructing outer domain boundaries and building a watertight and well connected CAD model. The 3D medial object algorithm also needs a certain level of quality from the input CAD model. Sharp corners, large vertex-face and edge-face gaps all need to be repaired before the medial object can be generated to guide the partitioning and ultimately the meshing. As the domain partitioning and meshing process respects the CAD topology, excessively short edges and narrow sliver faces should also be removed. The requirements outlined here are not that different to those imposed by standard surface and volume meshing algorithms, and typically can be automatically detected and removed.

4.2.2 Medial object

The medial axis, first introduced by Blum [7], is a method for analysing shapes. For a fluid domain, it can be defined as the set of all points in the domain which have more than one closest point on the boundary of the domain. If these points are taken together with their distance to the domain boundary (the *medial radius*), they form a complete description of the flow domain. The medial axis is computed and returned as a non-manifold CAD object called the medial object, which contains extra information to describe the relationships between the different components of the medial object along with medial radius information. See Fig. 4.2 for an example 3D medial object of the fluid domain around the NACA wing tip. The colour map used on the medial object indicates the distance from that point to the nearest (two) boundaries, namely the wing and the wall. The red line, on the other hand, represents the medial halo, described in the next section.

The medial object can be used for structured meshing, feature recognition and mid-surfacing as well as the automatic partitioning used here, and robust generation of the medial object has been a long standing challenge for the CAE[1] community. This algorithm is based on a domain Delaunay triangulation [69], and recent developments [10] allow it to robustly work on a range of production CAD models or in the air volume around such models.

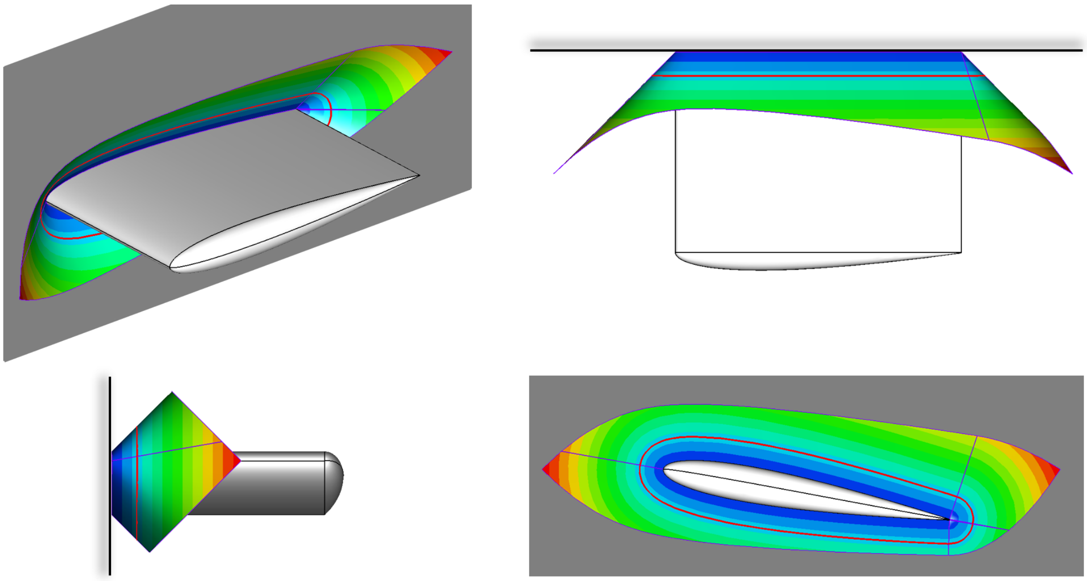


Figure 4.2: 3D medial object around the NACA wing tip.

[1] CAE: computer-aided engineering.

4.2.3 Partitioning

The 3D medial object is used to guide this partition generation in complex junctions. First the 3D medial object must be constructed, which is then used to generate an offset surface, or *shell*, from the boundaries of the fluid domain. The offset distance is defined directly by the user and will determine the thickness of the boundary layer mesh. The medial object is used to locate lines where simply offsetting the CAD faces would cause the shell to self-intersect, known as *medial halos* (the red lines in Fig. 4.2). The shell (Fig. 4.3) generated splits the fluid domain into two partitions: one near-field partition close to the boundary and one far-field partition. The near-field partition is subdivided into multiple smaller partitions using feature lines on the CAD model to guide the location of the partition faces.

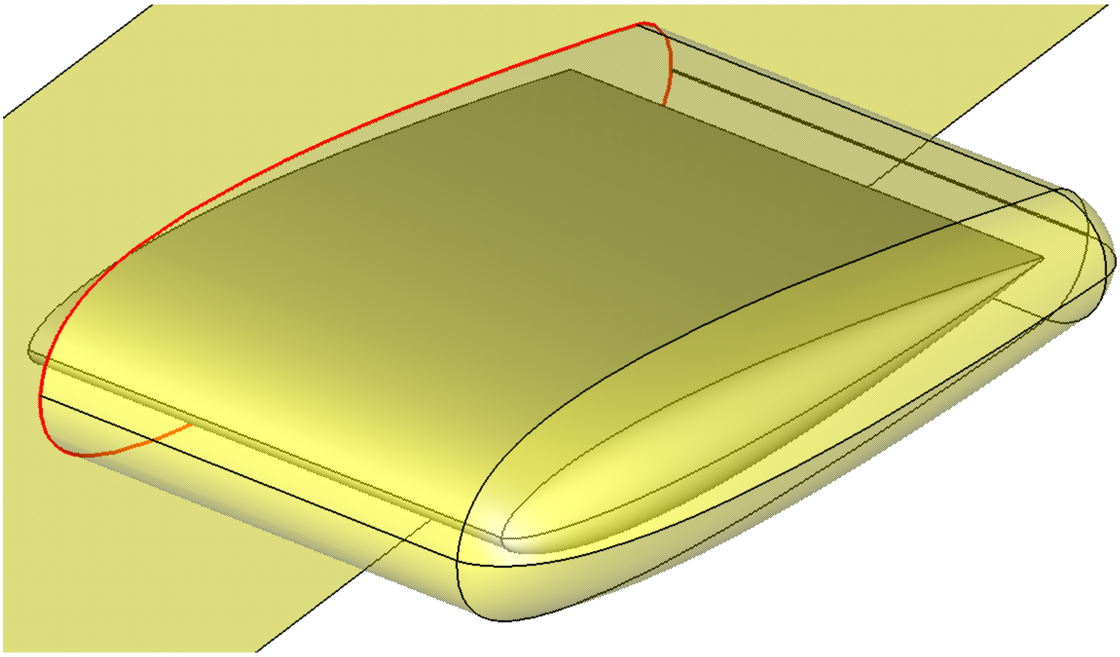
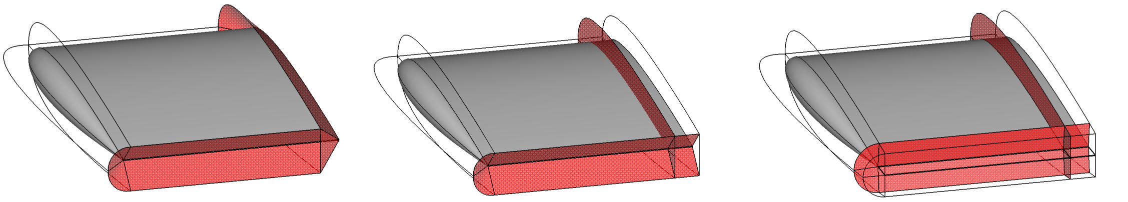


Figure 4.3: Shell around the NACA wing tip: medial halos in red.

If the fluid domain contains a sharp concave corner or edge (for example, at a wing/fuselage junction), flows will occur with potentially large velocity gradients in two or even three directions. Ideally this coarse linear mesh requires elements aligned with these principal directions. Using the medial object, there are several options available to achieve a mesh suitable for high-order upgrade.

The medial halos and medial object itself can be used to guide partition construction around concavities in wing root junctions, giving better mesh alignment when using hexahedral meshing. For streamlined bodies like a wing, three topologies are possible. They are shown in Fig. 4.4. The natural or direct topology that emerges from the medial axis is obtained when the medial object is used to separate different near-field blocks without any post-treatment of the concavities; see Fig. 4.4a. This topology, however, yields a distorted prismatic layer in orthogonal concavities, e.g. at wing-root junctions, as can be seen for example in Fig. 4.5. For this reason, a concave multi-normal topology (see Fig. 4.4b) is preferred. This topology requires the manipulation of the near-field partitions so as to obtain a hexahedral block at the junctions.



(a) Direct. (b) Concave multi-normal. (c) Concave/-vex multi-normal.

Figure 4.4: The different topologies and shell structures.

The use of a concave multi-normal topology, together with hexahedral elements, enhances the quality of the linear mesh at junctions. This is illustrated by Fig. 4.5, which compares the meshes obtained with a direct and a concave multi-normal approaches and shows that the concave multi-normal mesh avoids the distortion of prismatic layers at the concavity. The concave multi-normal mesh could potentially prevent the propagation of the prismatic layers due to self-intersection.

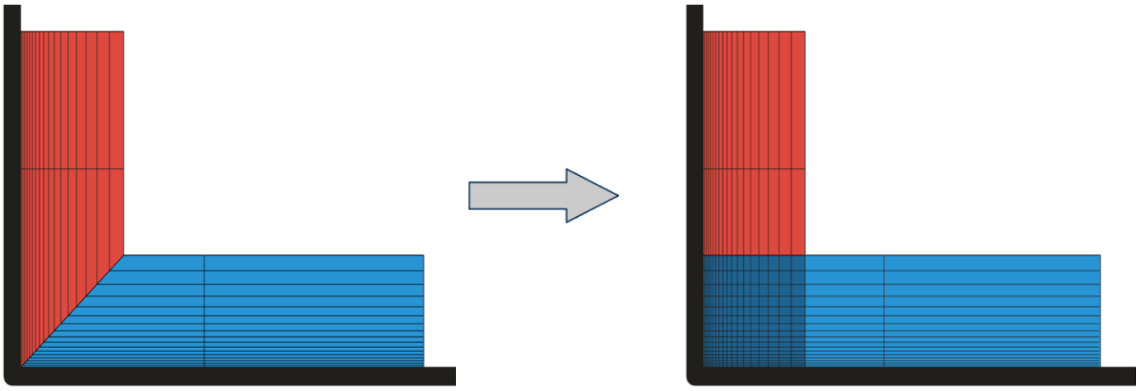


Figure 4.5: Meshes on direct and concave multi-normal topologies.

A third and last identified topology is the concave/convex multi-normal (see Fig. 4.4c), a further extension of the concave multi-normal topology. The concave/convex multi-normal topology is discussed further, in Sec. 4.5.

4.2.4 Linear mesh

The medial object based partitioning of the flow domain has been designed for use with certain mesh styles.

The concave multi-normal near-field topology permits the use of prismatic linear elements which can be swept from the CAD surfaces through the near-field partition to interface with a tetrahedral mesh in the far field, with only one element generated through the thickness of the near field partition.

A bottom-up mesh generation process is used to ensure the mesh is fully conformal between all partitions. Lines are meshed first, then surfaces are meshed with elements conforming to the lines, and finally the volumes of the partitions are meshed with elements which conform to the faces. The concave multi-normal topology features a structured hexahedral junction partition, and in this case the line meshes must be "balanced" to satisfy rules which are imposed via a structured mesh style. This is solved as an integer programming problem [73], and solved using an open source solver [26]. To further ensure good quality in the final mesh, a least-squares optimisation is performed to the line nodes to reduce potential skew.

The swept meshing of the partitions is performed by Delaunay triangulations of the designated template faces. This Delaunay mesh is swept into prismatic elements using the *CADfix* sweep mesher. Once the partition mesh and the tetrahedral far-field mesh have been completed, a mesh quality test is performed to make sure all elements produced during the linear meshing stage are not inverted. An example of a coarse linear mesh obtained with this method for the NACA wing tip geometry is shown in Fig. 4.6. The boundary layer mesh in the near-field region consists of 1,224 triangular prisms and 25 hexahedra and the far-field region is discretised into 12,576 tetrahedra. While a much finer mesh would be needed for simulations, these numbers show that the method is robust for coarse meshes too. In particular, the boundary layer mesh can be arbitrarily *a posteriori* refined to obtain any level of wall-normal resolution.

Periodicity

For the Rotor 67 example that follows in Sec. 4.4, a rotationally periodic mesh is required. The far-field boundary edges and surfaces need matching divisions so identical meshes can be made to ensure periodicity of the solution. This is achieved by performing an additional step in the linear mesh generation process during the balancing and alignment step. By calculating a rotational transformation which takes one side of the far field to the other, edges can be geometrically matched and the divisions copied from one side of the outer far field faces to the other.

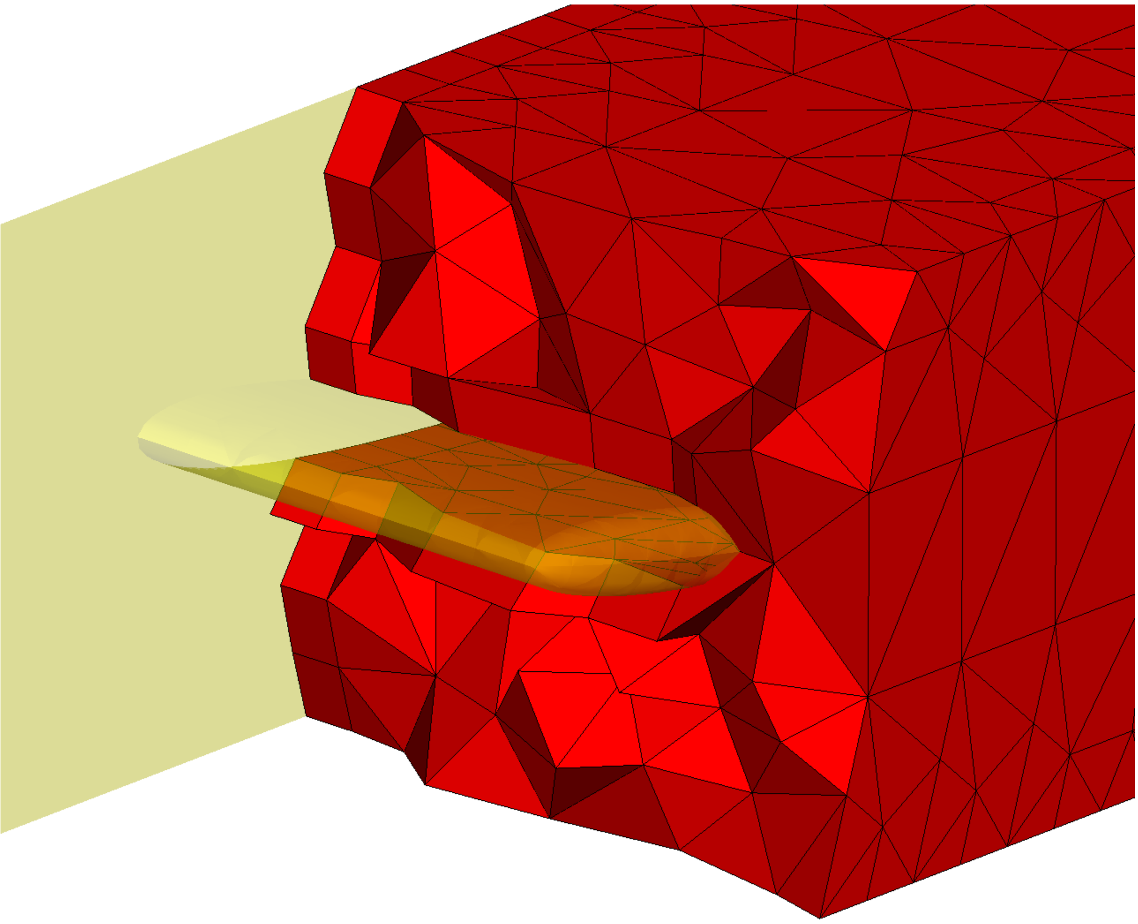


Figure 4.6: Linear mesh around the NACA wing tip.

As these edges and surfaces have already been balanced and aligned it is safe to duplicate the divisions on the other side of the far field, maintaining the density and quality required for these coarse meshes. Once the divisions have been duplicated the volume can be meshed and quality checked as outlined above.

4.3 High-order mesh

The *a posteriori* generation of a high-order mesh from a linear mesh proceeds in a bottom-up fashion following the ideas proposed in reference [70]. The additional points required for the high-order polynomial discretisation are incorporated sequentially first along the curves, then on the surfaces of the CAD geometry and, finally, in the interior of the domain. The generation of points along the curves is essentially the one proposed in reference [70]. The following sections describe the improvements incorporated into the methodology to achieve the type of meshes sought in this work.

4.3.1 Optimisation of the surface mesh

Inaccuracies in the representation of the geometry of the boundary of the computational domain due to CAD distortion, even if small, could have a significant impact on the accuracy of the flow solution. To overcome this problem, *NekMesh* optimises the location of the high-order nodes in the mesh to reduce distortion by modelling the mesh entities as spring networks and minimising their deformation energy. The optimal location of the mesh nodes is obtained in a bottom-up fashion.

The first step is to optimise the location of mesh nodes belonging to edges that lie on CAD curves by minimizing the deformation energy of a spring system in the parametric space of the curve with the vertices in the linear mesh fixed. This is followed by the processing of mesh nodes on edges that lie on the CAD surfaces. Again, the optimal position of the mesh nodes of an edge is found on the 2D parametric space of the surface by minimizing the deformation energy of the 3D high-order edge. As a result, the optimised high-order edge will lie approximately on the geodesic between the two end points on the surface [70]. The final step is the relocation of the mesh nodes in interior triangle faces that lie on CAD surfaces. Here, the mesh nodes on the edges of the triangles are fixed and the interior nodes are free to move. Each of the free interior nodes is connected to a system of six surrounding nodes by springs. The minimum deformation energy of this system of spring is found using a bounded version of the BFGS[2] algorithm that accounts for the limits of the parameter space in the CAD entities [16]. The gradients required by the optimisation procedure can be evaluated from the CAD information provided by *CFI*. This procedure leads to an overall distribution of points which ensures the surface mesh is smooth, unless pathological distortion is present in the CAD geometry.

4.3.2 Refined boundary layer mesh

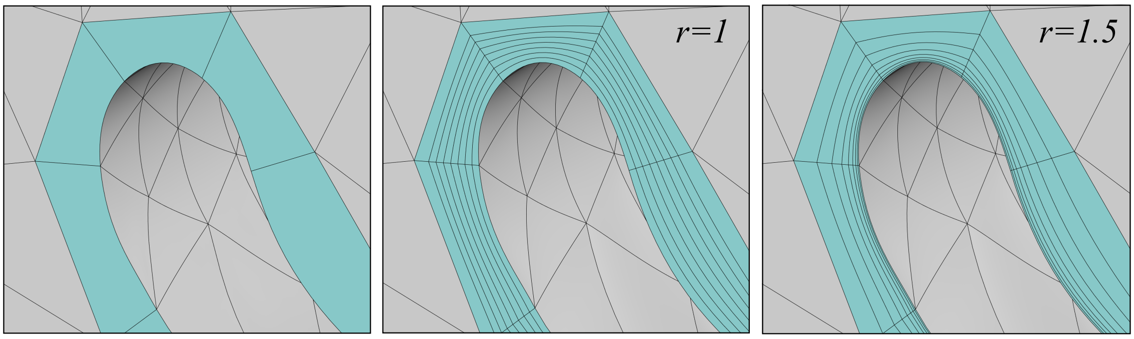
The generation of highly stretched elements with high aspect ratios, say 100:1, which are required to accurately simulate the high shear of boundary layer flows at aeronautically-relevant Reynolds numbers, poses a significant challenge for high-order mesh generation. If the high-order mesh is produced using *a posteriori* methods, then curving thin elements in the boundary layer mesh will almost certainly produce self-intersecting elements in regions of high curvature. To avoid this, high-order boundary layer meshes are generated by applying the isoparametric approach [54] to the linear meshes produced via the medial object.

First a macro boundary layer hybrid mesh consisting of a single layer of hexa-

[2] BFGS: Broyden–Fletcher–Goldfarb–Shanno.

hedra and triangular prisms is generated by the medial object method in the near-field region and the far-field partition is discretised into tetrahedra. The medial object allows the thickness of the near-field region and the height of the elements to be controlled to a much greater extent than most commercial mesh generators. By selecting a thickness of the shell that gives enough room to accommodate the surface curvature, the likelihood of generating invalid high-order elements within the macro boundary layer mesh is reduced.

The volume generation proceeds next to split these high-order elements using the isoparametric approach [54], previously described in Sec. 3.2.5. The splitting strategy used here is to specify a number of subdivisions, or layers, along the parametric coordinate representing the wall normal, and a growth, or progression, rate for the height of the elements. This progression rate is characterised by a factor r that is the ratio of heights of adjacent elements. The generation of the boundary layer mesh by this approach is illustrated in Fig. 4.7, which shows a boundary layer region of macro-prisms split to produce highly curved, valid boundary layer elements with very high aspect ratio.



(a) Coarse mesh. (b) Uniform split. (c) Non-uniform split.

Figure 4.7: Boundary layer meshes using the isoparametric splitting: adapted from reference [54].

Bidirectional isoparametric splitting

The generation of concave multi-normal meshes in junction regions allows for the generation of structured, hexahedral elements that avoid the effects of layer distortion. However, this approach breaks the original isoparametric splitting of prismatic stacks of elements that has been used thus far to produce boundary layer meshes of arbitrary thickness. The method is adapted to deal with concave multi-normal boundary layer refinement in order to generate valid, curved meshes through a straightforward extension of the technique.

A similar extension to bidirectional splitting was described in Sec. 3.2.5 for two-dimensional quadrilaterals. The extension of this technique to three-dimensional hexahedra is straightforward, so much so that tridimensional splitting could even be achieved. An example of the boundary layer mesh generated by bidirectional hexahedral splitting for the wing tip geometry is shown in Fig. 4.8. The mesh in the near-field region has been subdivided into 10 layers using a growth rate $r = 1.5$. The resulting boundary layer mesh is formed by 12,240 prismatic and 2,500 hexahedral elements with an aspect ratio of up to 70.

4.3.3 Volume mesh

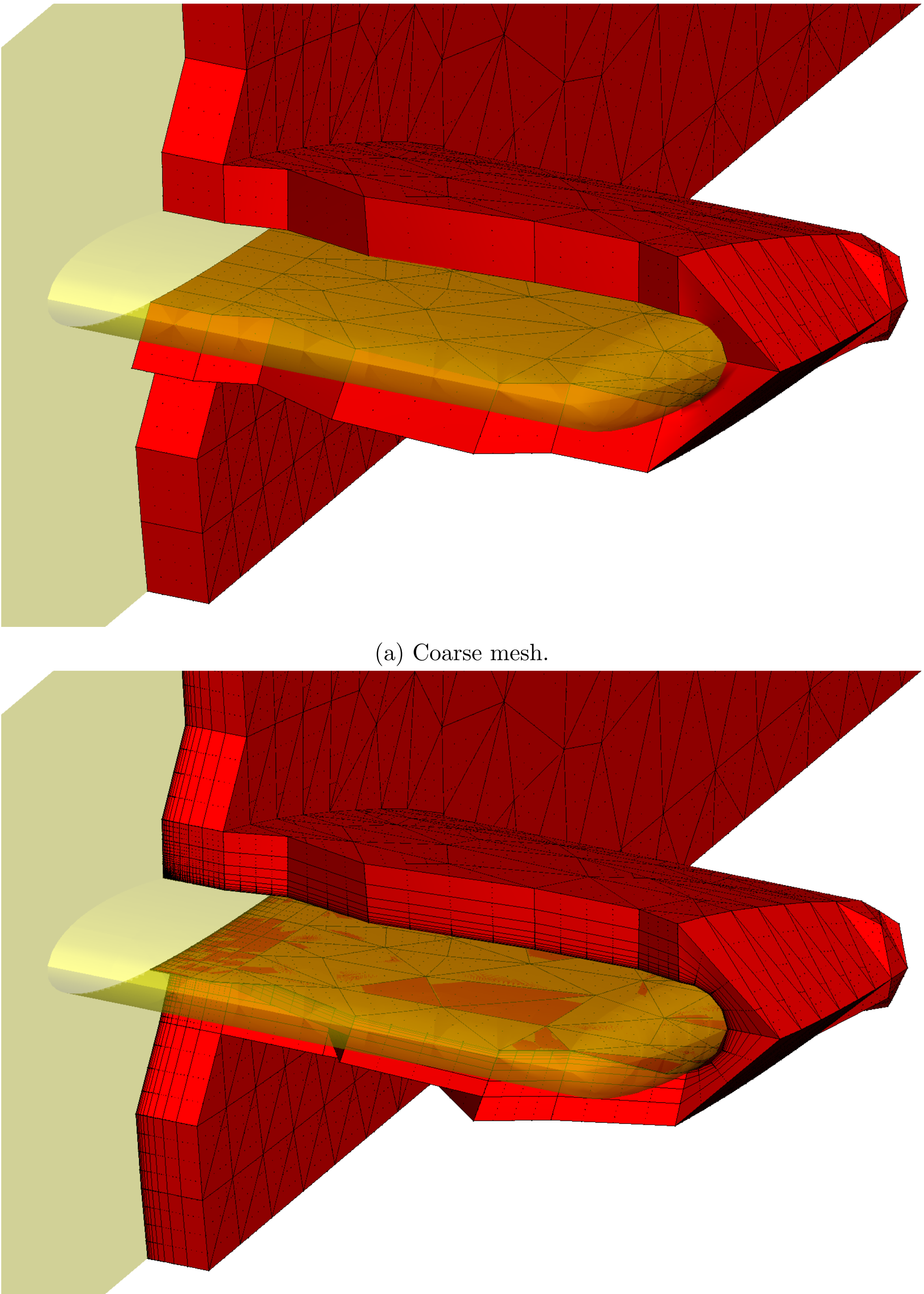
The introduction of the curvature of the CAD surfaces onto the high-order surface triangulation produces high-order elements in the interior of the volume with curved faces and edges that could become invalid. Controlling the thickness of the near-field via the medial object permits the generation of linear boundary layer meshes that can accommodate the deformation induced by surface curvature without producing invalid elements. The positions of the additional nodes required for the polynomial representation of the high-order elements are obtained by means of a mapping between a reference element and the physical element which accounts for the presence of curvature on its faces and edges lying on the CAD definition, whilst the other edges are straight and their faces planar.

A high-order mesh of the volume is shown in Fig. 4.9 which contains, in addition to the elements in the boundary layer mesh, 12,576 tetrahedral elements of polynomial order 4.

4.4 Application

This section presents an illustration of the proposed mesh generation methodology and the high-order meshes it produces using two geometries proposed by NASA[3] for CFD validation: the Common Research Model and the Rotor 67. These are described in the following sections and, for reference, all the corresponding high-order meshes have been generated using a polynomial order of 4.

[3] NASA: National Aeronautics and Space Administration.



(a) Coarse mesh.

(b) Split mesh.

Figure 4.8: High-order boundary layer meshes around the NACA wing tip: the wing and wall surfaces are shown in yellow for perspective.

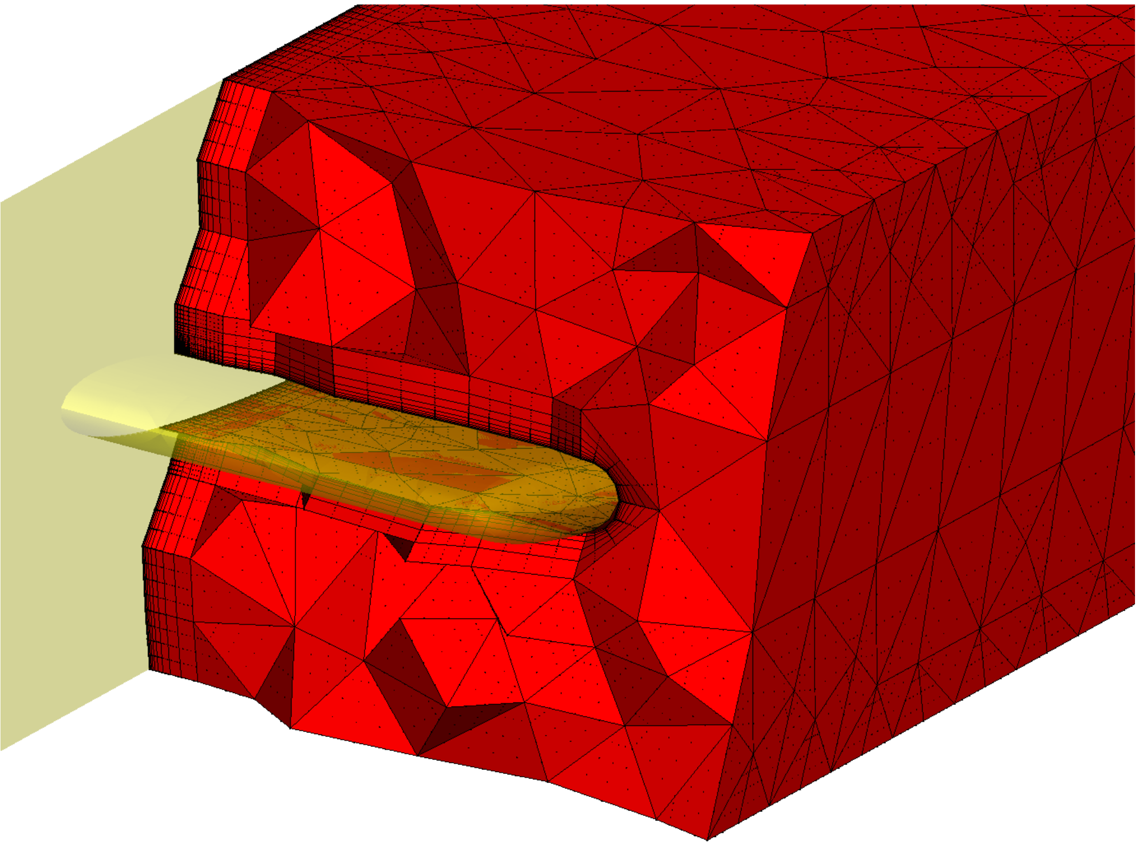


Figure 4.9: Final high-order mesh around the NACA wing tip.

4.4.1 NASA Common Research Model

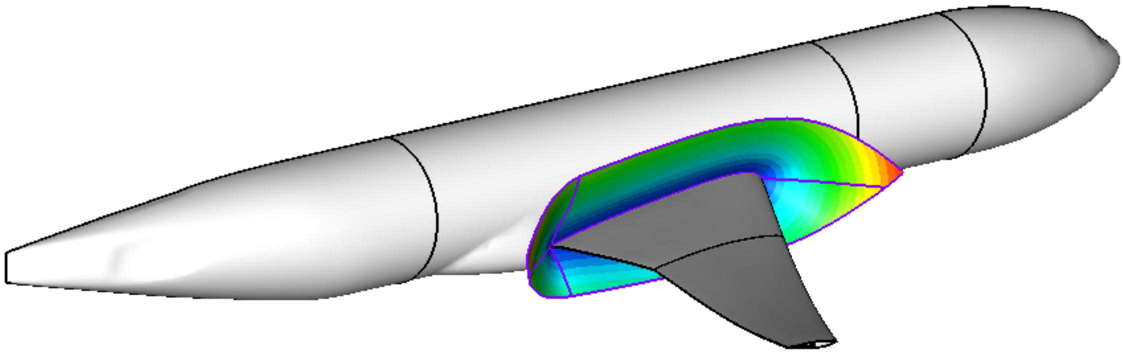
The NASA CRM[4] presented here is one of the five configurations designed by NASA [81] for CFD validation. It is a wing/body alone configuration with a fuselage with a maximum radius of 0.17m, and a 35 degrees backward-swept wing of aspect ratio of 9 and span of 1.60m.

In the first instance, the original definition of the CAD geometry in STEP[5] format [57] is processed through *CADfix* to clean it and fix a number of inconsistencies and severe distortions that might prevent the successful generation of the medial object decomposition and the high-order mesh.

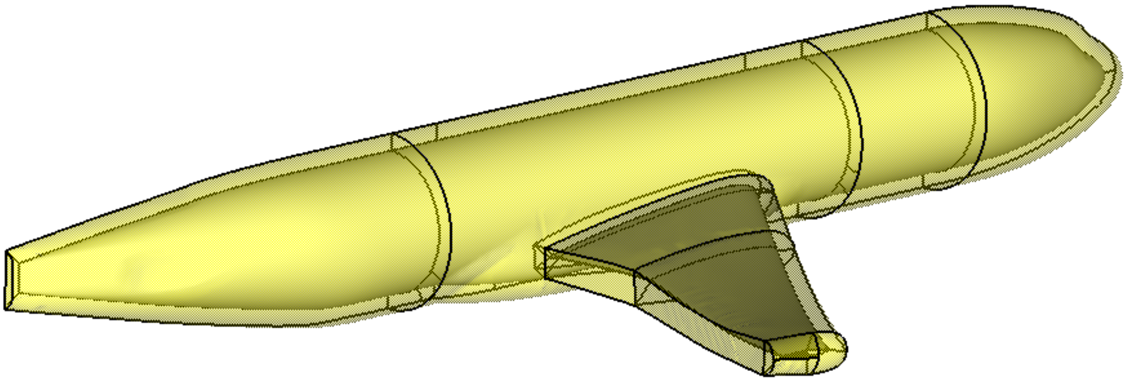
The medial object interface, depicted in Fig. 4.10a, was designed to generate a hexahedral mesh at the wing-fuselage junction. Fig. 4.10b shows the block partitioning of the near-field region that provides the framework for generating the boundary layer mesh around the aircraft. Fig. 4.11 provides a more detailed view of the blocks in the near-field region via a wireframe representation of the edges of the partitions in that region.

[4] CRM: Common Research Model.

[5] STEP: Standard for the Exchange of Product model data; ISO 10303.



(a) Interface of the medial object at the junction.



(b) Shell.

Figure 4.10: Medial object around the NASA CRM.

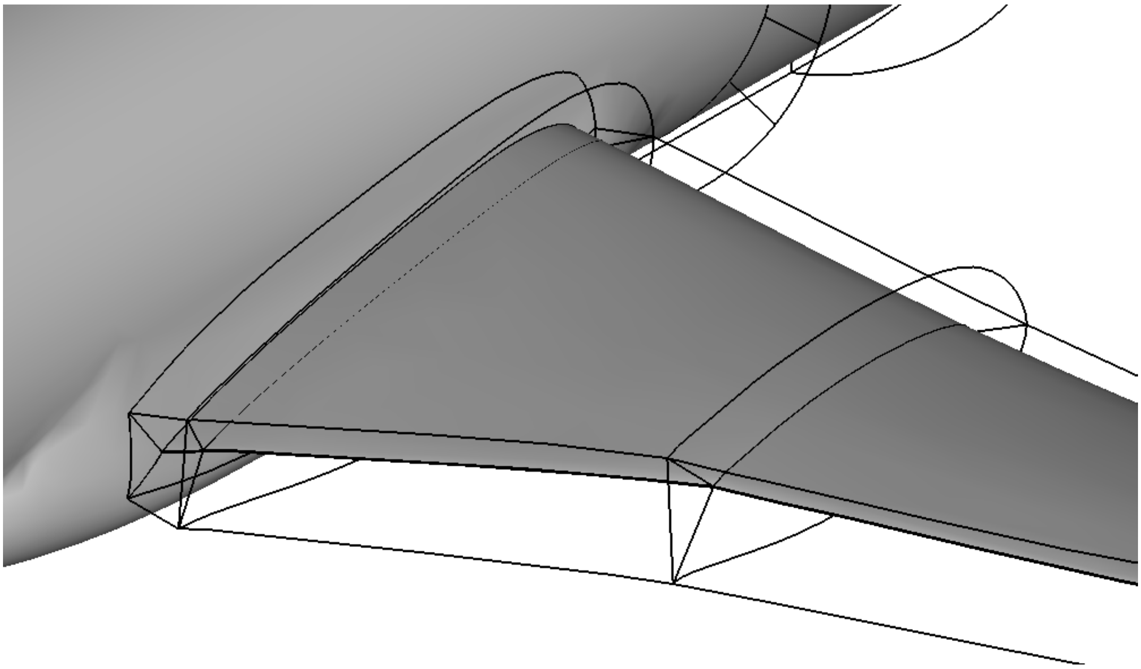
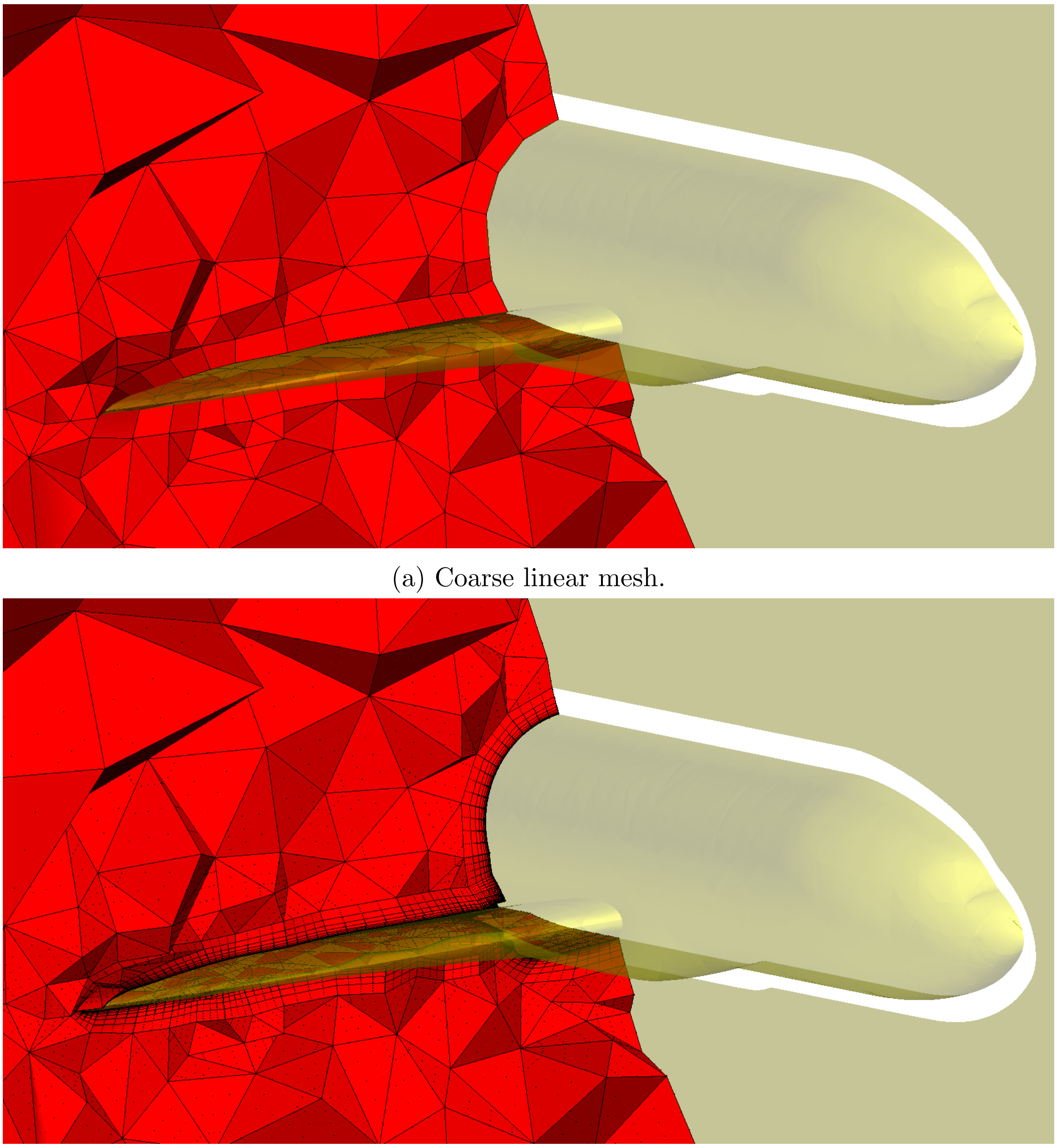


Figure 4.11: Wireframe of the partitions around of the NASA CRM: close-up near the junction..

The medial object decomposition was used to produce an initial coarse linear mesh with a single layer of elements in the near-field partition. This boundary layer mesh consisted of 33 hexahedra and 2,042 prisms. The far-field region was discretised using 18,084 tetrahedra. The characteristics of the linear mesh can be observed in Fig. 4.12a that shows a cut normal to the fuselage through the mesh and in the enlargement of that mesh in the wing-fuselage junction of Fig. 4.13a.



(a) Coarse linear mesh.

(b) Split high-order mesh.

Figure 4.12: Volume mesh around the NASA CRM: cut view.

In the final step of the generation process the isoparametric approach is applied to subdivide the coarse boundary layer mesh into 10 layers with a progression ratio

$r = 1.5$. This produces a high-order mesh with 20,420 prisms and 3,300 hexahedra with a maximum element stretching ratio of 60. Views of the cut through the high-order mesh and the enlargement near the wing-fuselage junction are shown in Fig. 4.12b and Fig. 4.13b, respectively.

4.4.2 NASA Rotor 67

The geometry considered here is a first stage rotor (NASA Rotor 67) of a two-stage transonic fan designed and tested at the NASA Glenn center [61]. The original rotor has 22 blades with a tip leading-edge radius of 25.7cm and a tip trailing-edge radius of 24.25cm. The hub to tip radius ratio is 0.375 at the leading edge and 0.478 at the trailing edge. Here the geometry for a single blade with periodic BC and without tip clearance is considered. The distinctive feature of this geometry is the incorporation of periodic features both in the medial object decomposition and the linear and high-order meshes.

The medial object for this geometry is shown in Fig. 4.14a. Fig. 4.14b depicts the block decomposition in the near-field region. The linear mesh in that region, which consists of 53,830 prisms and 236 hexahedra, is shown in Fig. 4.14c.

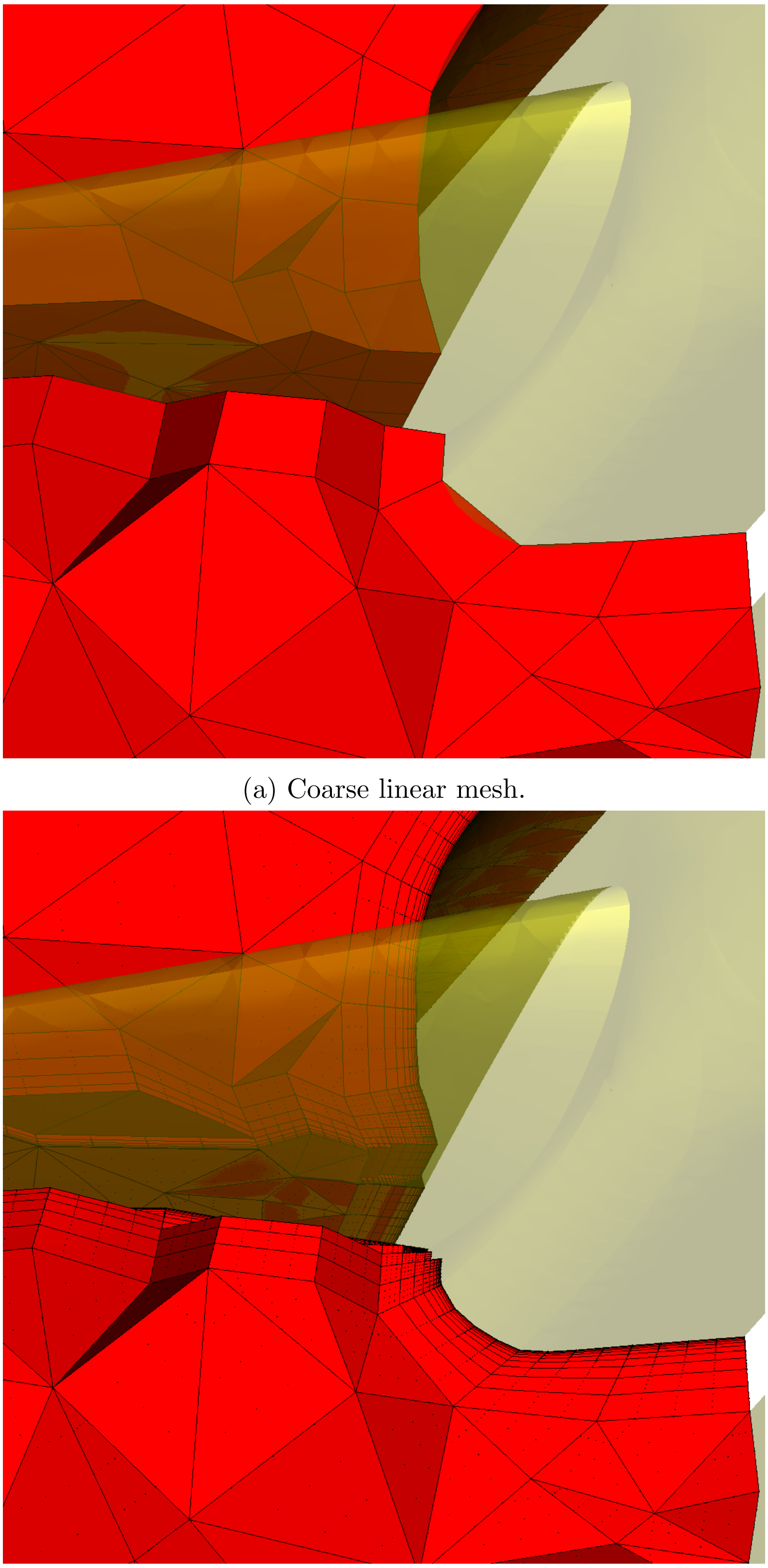
Enlarged views of the surface mesh and the boundary layer mesh in the vicinity of the blade root are shown in Fig. 4.15a and Fig. 4.15b, respectively. The surface mesh contains 53,830 triangles and 472 quadrilaterals.

The coarse mixed prismatic-hexahedral linear mesh is split via the isoparametric mapping into 10 layers with a progression ratio $r = 1.5$ which leads to a maximum stretching ratio of 80 for the elements near the wall. This linear mesh is then transformed into a high-order mesh with polynomial order four. A close-up view of the high-order surface mesh near the leading edge of the blade's root is shown in Fig. 4.16.

A pictorial summary of the mesh characteristics of the linear and high-order meshes is given in Fig. 4.17 which shows enlargements of these meshes around the mid-chord of the blade in the near-field region.

4.5 Concave/convex multi-normal topologies

This approach can be extended to concave/convex multi-normal topologies (see Fig. 4.4c) too. This topology is typical of streamlined bodies with a sharp or squared-off trailing edge, the former being a specific case with zero thickness. In this configuration, the near-field blocks are extended downstream instead of looping around the trailing edge. This makes it possible to obtain a structured mesh in the



(a) Coarse linear mesh.

(b) Split high-order mesh.

Figure 4.13: Volume mesh around the NASA CRM: cut, close-up view on the junction.

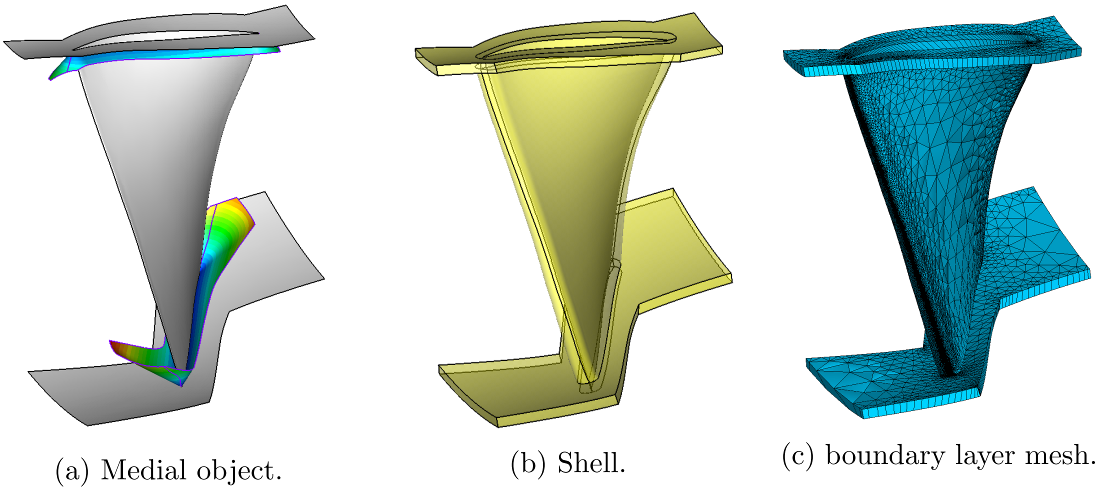
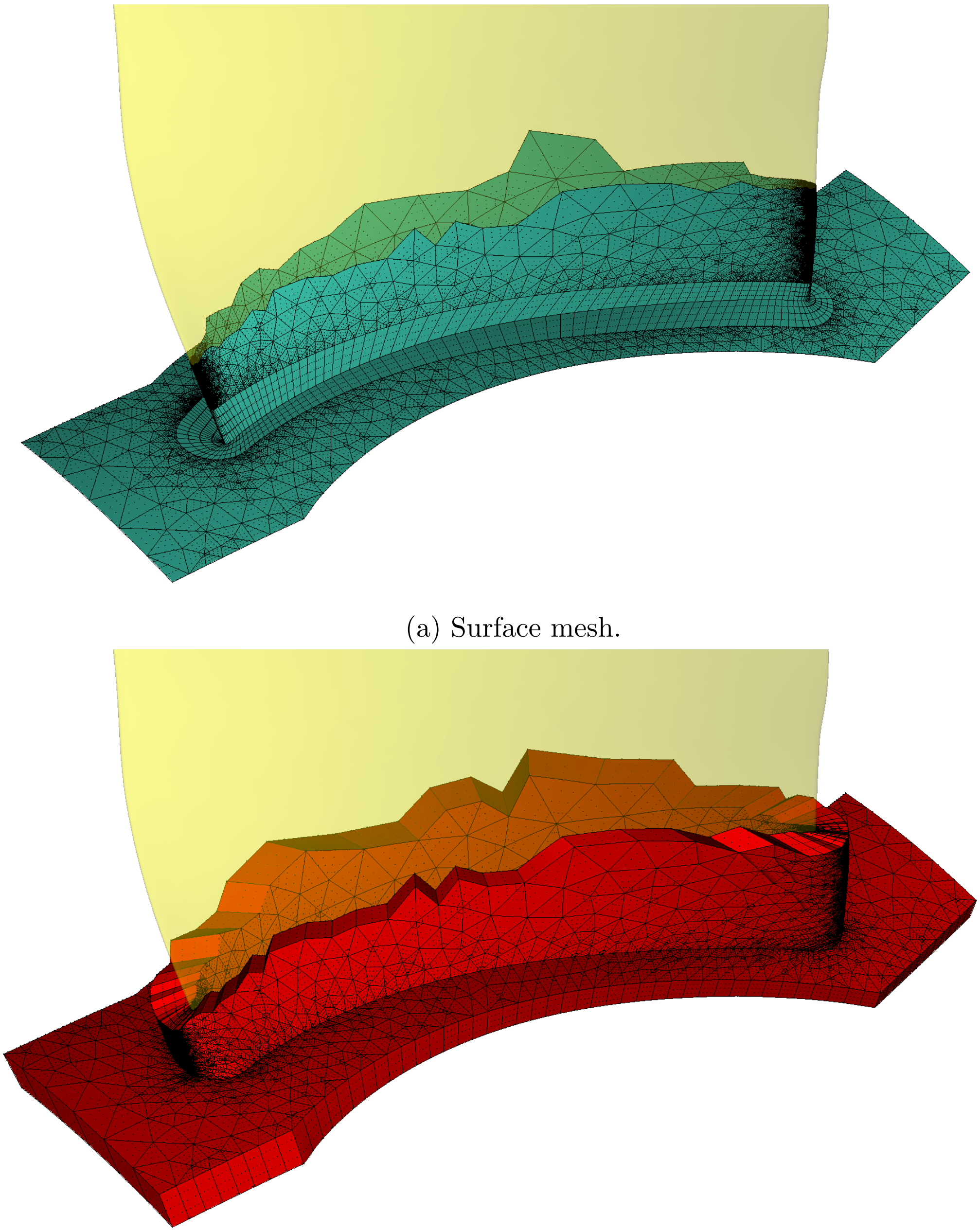

(a) Medial object. (b) Shell. (c) boundary layer mesh.

Figure 4.14: Near-field partition around the NASA Rotor 67.

wake of the body too. While the boundary layer blocks were previously closed in direct and concave multi-normal configurations, they are now open in concave/convex multi-normal configurations. The open end of the blocks are exposed to the far-field unstructured mesh and splitting them will require the propagation of these subdivisions into the unstructured mesh.

The original workflow is modified to accommodate for the concave/convex multi-normal topology. During the generation of the linear mesh, only the coarse boundary layer mesh in the near-field partitions is produced; not the far-field unstructured mesh just yet. The partial linear mesh is passed on to *NekMesh* where a high-order mesh is produced and split in the exact same way as for direct and concave multi-normal topologies. Improvement of the data flow in *NekMesh* had to be made. In particular, the conservation of CAD information within the data structures of *NekMesh* is important both for nodes and for elements of all dimensionality so that the future export (see below) to *CADfix* and the rest of the workflow can be carried out. Not only did the CAD information have to flow throughout the *NekMesh* workflow, new nodes and elements are also created during the isoparametric splitting that need to be matched with the relevant CAD entities.

After splitting, the mesh is finally returned to *CADfix*. A new module had to be implemented to cover this export in a way that mirrors the initial import of a *CADfix* mesh into *NekMesh*. Importantly, nodes and elements must be exported to *CADfix* with the right CAD information so that the surface meshes on the boundaries of the far-field partitions can be appropriately extracted. From this surface mesh, the unstructured mesh can finally be generated in *CADfix* with the knowledge of the subdivisions of the boundary layer mesh. Because the boundary layer mesh is open-ended downstream, a quadrilateral interface exists between the near- and the



(a) Surface mesh.

(b) boundary layer mesh.

Figure 4.15: Near-field partition around the NASA Rotor 67: close-up view on one junction.

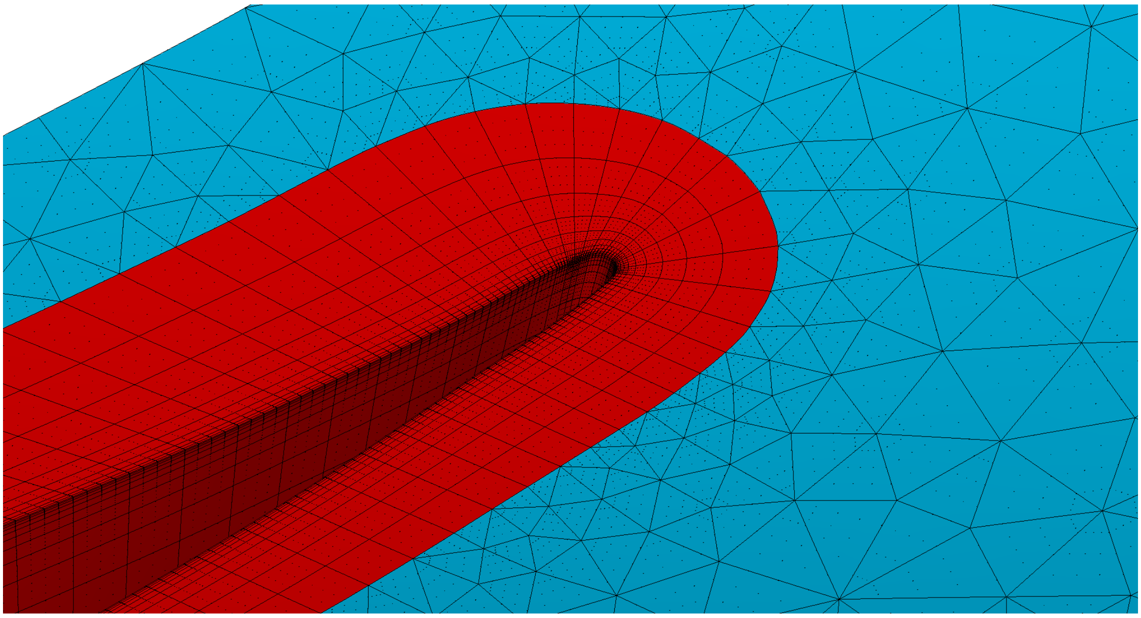
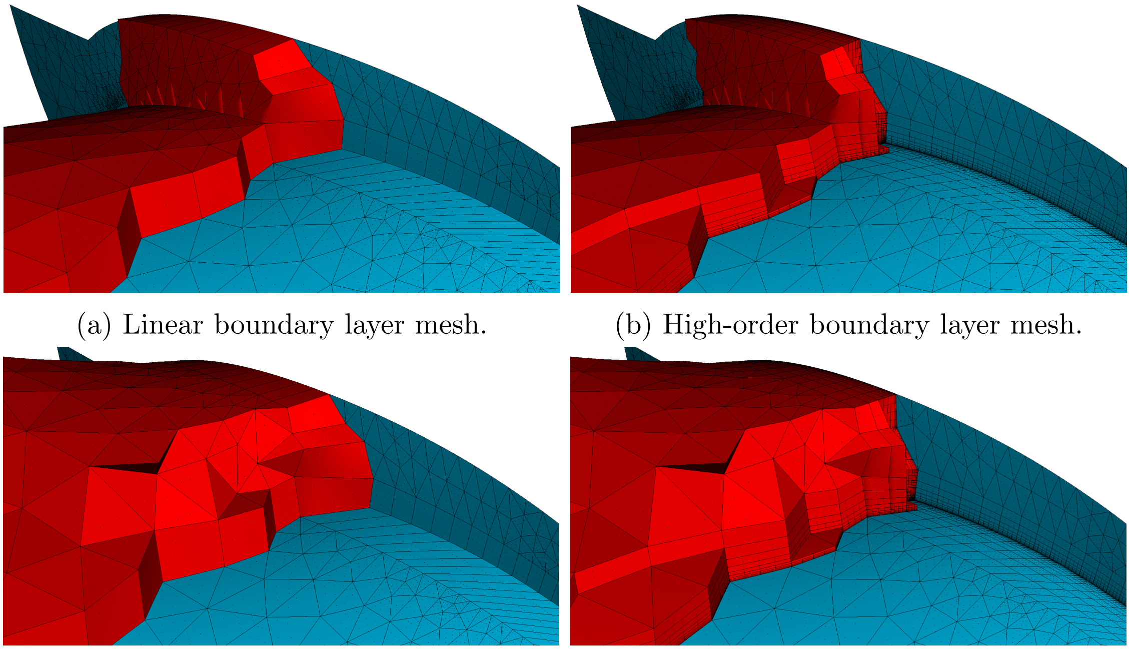


Figure 4.16: Split surface mesh around the NASA Rotor 67: close-up view on the leading edge.



(a) Linear boundary layer mesh.

(b) High-order boundary layer mesh.

(c) Linear volume mesh.

(d) High-order volume mesh.

Figure 4.17: Meshes around the NASA Rotor 67: close-up view on the mid-chord.

far-field meshes, which must be connected with pyramids before transitioning to an all-tetrahedra mesh.

Difficulty can, however, arise in the extension of the wake partition when the surface to which the body is fixed is not plane, e.g. in a wing-fuselage configuration. In this situation, the wake-fuselage junction not only would be hard to trace along the curved fuselage but it might also expand out of reach in either or both the streamwise and the spanwise directions. For this reason, a gap may be introduced between the boundary layer mesh generated around the fuselage and the wake block. This approach gives greater flexibility in the generation and manipulation of the wake block partition. This is illustrated in Fig. 4.18 on the NACA wing tip. This wing has a squared-off trailing edge of finite thickness and is attached to a flat wall. In this example, the wake block has been expanded on purpose to demonstrate the flexibility of the linear mesher to generate wake blocks suited to the flow configuration.

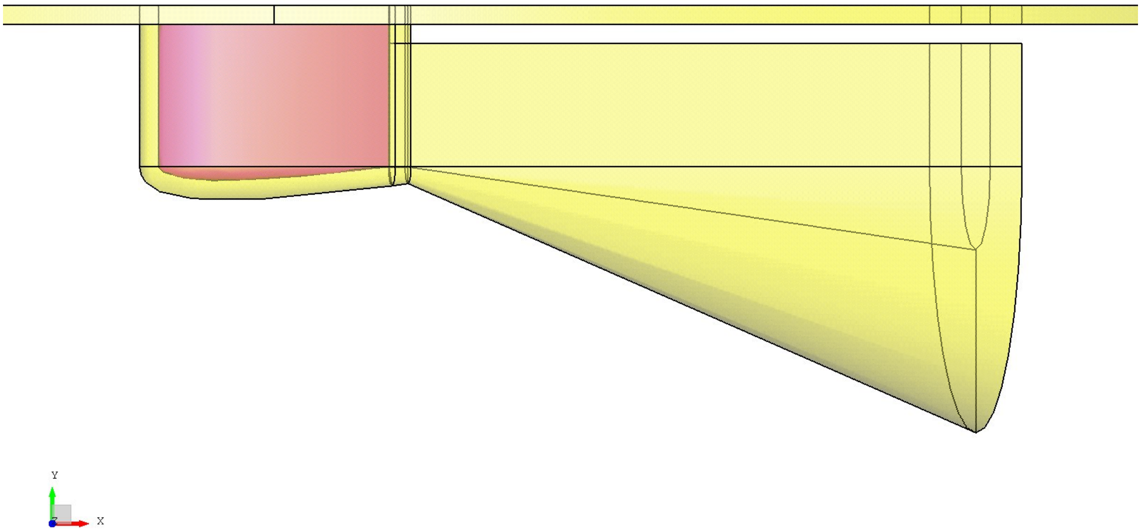
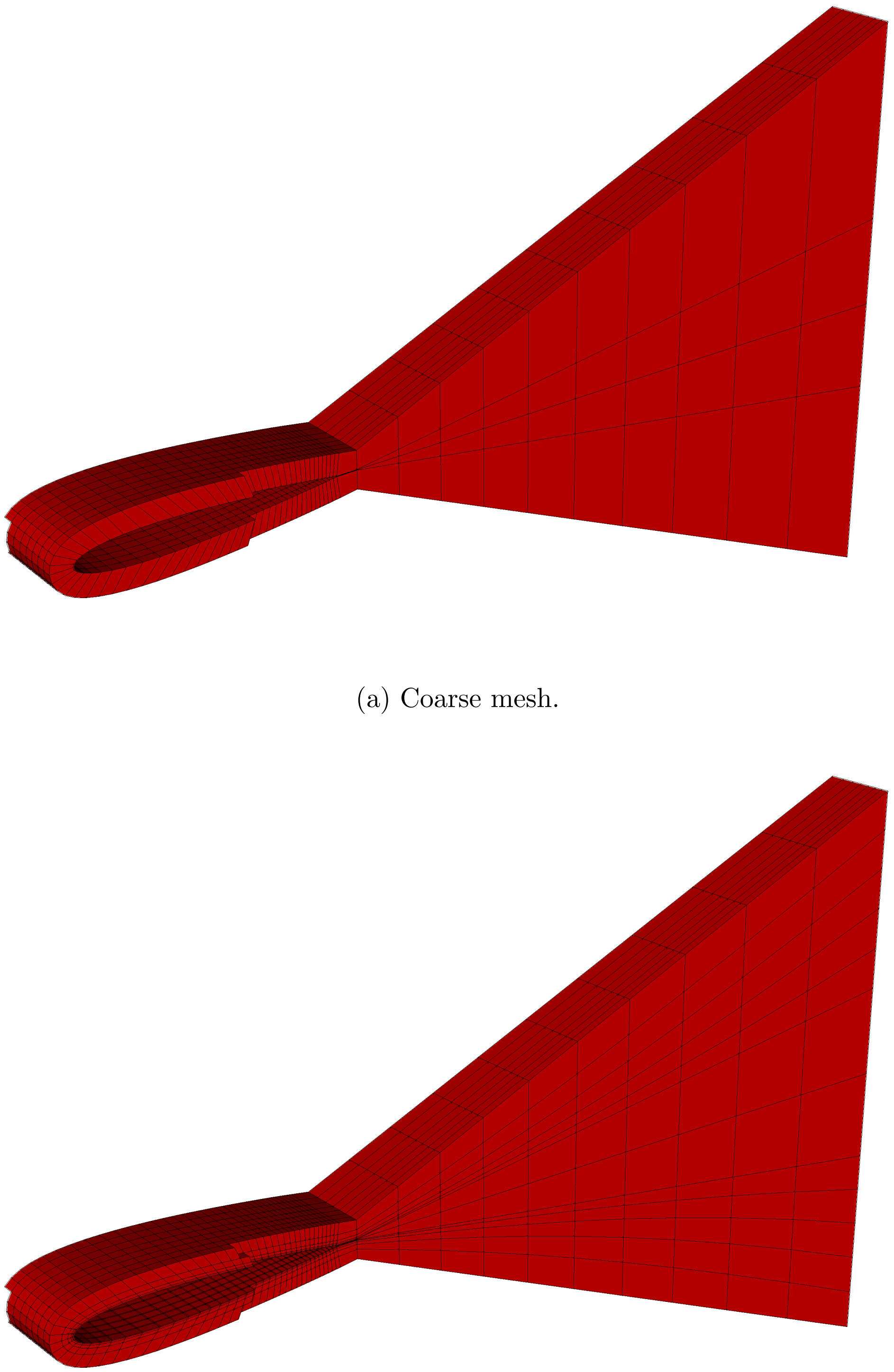


Figure 4.18: Gap between the wake block of the NACA wing tip and the wall.

The procedure then proceeds. The near-field is discretised with a linear mesh, then transferred to *NekMesh* for high-order meshing and finally split. The structured mesh on a portion of span before and after splitting is shown in Fig. 4.19. A progression ratio of 2.0 and 5 elements in the thickness were used for the splitting in the boundary layer of the wing. The progression then linearly varies with distance from the trailing edge until reaching 1.0 (uniform distribution of elements) at the far end of the wake.



(a) Coarse mesh.

(b) Split mesh.

Figure 4.19: Near-field meshes around the NACA wing tip with concave/convex multi-normal topology.

Chapter 5

r-adaptation

In r-adaptation the aim is to increase resolution by locally reducing the mesh size, h, whilst keeping the number of DOF in the mesh constant. This effectively requires mesh nodes to be clustered in the vicinity of those regions where additional resolution is required, e.g. at shocks. This chapter proposes to accomplish this by adapting a variational framework for the optimisation of high-order meshes [80].

First, the variational formulation for high-order mesh optimisation is recalled in Sec. 5.1. The proposed modification of this technique for r-adaptation is then described in Sec. 5.2 along with some analytical examples. Sec. 5.3 finally proposes to use an error estimator to drive the adaptation process for compressible flow simulations.

5.1 Variational mesh optimisation

The objective of this variational framework [80] is to improve the quality of high-order *curvilinear* elements using a node-based optimisation approach using a formulation based on the energy of deformation. An important aspect of such energy-based formulation is that a suitable choice of the energy functional, namely polyconvex, would guarantee the existence of a minimum and therefore of a solution to the minimisation problem.

Fig. 5.1 shows that a mapping $\boldsymbol{\phi}_M$ exists from a reference element $\boldsymbol{\Omega}_{st}$ to a curvilinear high-order element $\boldsymbol{\Omega}^e$. The mapping $\boldsymbol{\phi}_M$ can be further decomposed into two distinct mappings: a mapping $\boldsymbol{\phi}_I$ from reference to *ideal* elements and a mapping $\boldsymbol{\phi}$ from the *ideal* to the curvilinear elements. The ideal element is defined as the high-order linear element, which after minimisation will be the element that the optimiser seeks to achieve.

From this ideal element, the deformation energy is calculated. The mesh is

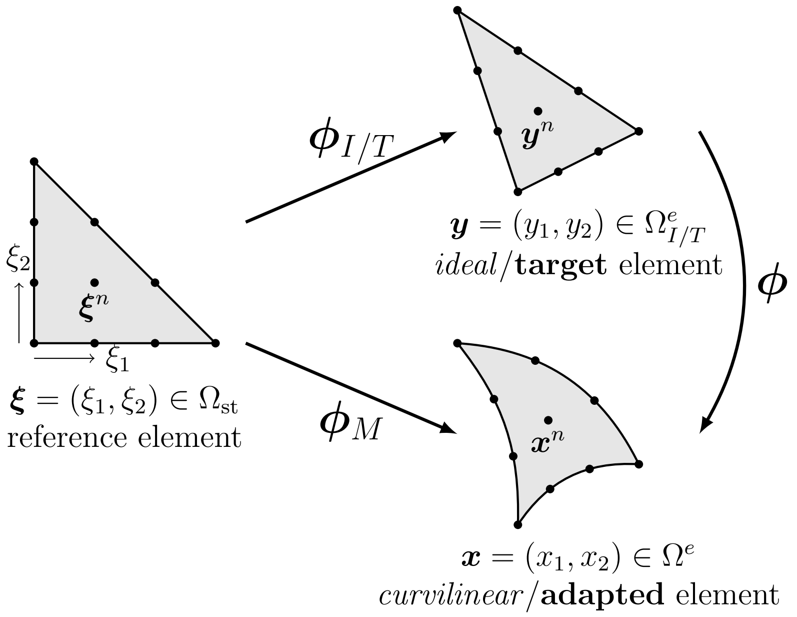

$\phi_{I/T}$
y^n
$\boldsymbol{y} = (y_1, y_2) \in \Omega^e_{I/T}$
ideal/**target** element
ϕ
ξ_2
ξ^n
ξ_1
$\boldsymbol{\xi} = (\xi_1, \xi_2) \in \Omega_{\text{st}}$
reference element
ϕ_M
x^n
$\boldsymbol{x} = (x_1, x_2) \in \Omega^e$
curvilinear/**adapted** element

Figure 5.1: Mappings between the reference, the *ideal*/**target** and the *curvilinear*/**adapted** elements.

deformed to minimise an energy functional $\mathcal{E}(\nabla\boldsymbol{\phi})$:

$$\text{find} \min_{\boldsymbol{\phi}} \mathcal{E}(\nabla\boldsymbol{\phi}) = \int_{\boldsymbol{\Omega}^e} W(\nabla\boldsymbol{\phi}) d\boldsymbol{y}, \tag{5.1}$$

where $W(\nabla\boldsymbol{\phi})$ is a formulation of the deformation energy. Several formulations were tested and it was found that the best results were obtained when using a hyperelastic model [80]. For this model, the strain energy takes the form

$$W = \frac{\mu}{2}(I_1^{\boldsymbol{C}} - 3) - \mu \ln\ J + \frac{\lambda}{2}(\ln\ J)^2, \tag{5.2}$$

where λ and μ are material constants, $\boldsymbol{C}$ is the right Cauchy-Green tensor, $I_1^{\boldsymbol{C}}$ is its trace and J is the determinant of the Jacobian matrix $\boldsymbol{J} = \nabla\boldsymbol{\phi}$. This formulation is used in the work that follows.

When optimising a high-order mesh, the ideal element of Fig. 5.1 corresponds to the linear element before generation of a high-order element by addition and projection of high-order nodes. To obtain best performance, not only high-order nodes but vertices are optimised too. This ideal element becomes an arbitrary **target** element. The optimiser now aims at **adapting** element $\boldsymbol{\Omega}^e$ towards a size and a shape similar to the target element $\boldsymbol{\Omega}^e_T$.

5.2 $\boldsymbol{r}$-adaptation

In the framework of adaptive meshes, the ideal element $\boldsymbol{\Omega}_I^e$ becomes a target element $\boldsymbol{\Omega}_T^e$. Because this target element $\boldsymbol{\Omega}_T^e$ is modified, the optimisation of the mesh by minimisation of the energy functional $\mathcal{E}(\nabla\boldsymbol{\phi})$ will force element $\boldsymbol{\Omega}^e$ towards a shape and dimensions similar to $\boldsymbol{\Omega}_T^e$. The manipulation of $\boldsymbol{\Omega}_T^e$ is achieved by modification of its mapping $\boldsymbol{\phi}_T$ and can be isotropic or anisotropic alike. Linear transformations can be applied to the Jacobian of the mapping $\boldsymbol{J}_T = \nabla\boldsymbol{\phi}^T$. In the anisotropic case, the Jacobian is multiplied by a metric tensor $\boldsymbol{M}$:

$$\boldsymbol{J}_T = \boldsymbol{M}\boldsymbol{J}_I \tag{5.3}$$

In the isotropic case, the Jacobian is simply scaled by a linear factor r, which can in turn be expressed more generally as a metric tensor $r\boldsymbol{I}$ multiplication:

$$\boldsymbol{J}_T = r\boldsymbol{J}_I = (r\boldsymbol{I})\boldsymbol{J}_I \tag{5.4}$$

5.2.1 Application

This section demonstrates the feasibility of this method using analytical expressions for the metric tensor. The example hereby presented proposes to adapt a homogeneously meshed double unit side domain such as the one shown in Fig. 5.2a. The aim is to refine along the circumference of a circle of unit diameter. This is achieved anisotropically by shrinking elements in the radial direction only.

A scaling factor r is defined in the radial direction at an angle α from the x-axis. The metric tensor can be expressed as a succession of linear 2D transformations:

1. Rotate the element so that the radial axis coincides with the x-axis

$$\boldsymbol{M}_1 = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \tag{5.5}$$

2. Scale the element along the x-axis

$$\boldsymbol{M}_2 = \begin{pmatrix} r & 0 \\ 0 & 1 \end{pmatrix} \tag{5.6}$$

3. Rotate the element back to its initial orientation

$$\boldsymbol{M}_3 = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \tag{5.7}$$

The combined metric tensor becomes:

$$\boldsymbol{M} = \boldsymbol{M}_1\boldsymbol{M}_2\boldsymbol{M}_3 = \begin{pmatrix} 1 - s\cos^2\alpha & s\sin\alpha\cos\alpha \\ s\sin\alpha\cos\alpha & 1 - s\sin^2\alpha \end{pmatrix}, \tag{5.8}$$

where $s = 1 - r$ is the relative reduction in size.

Additionally, a distribution of $r(d)$ is defined, with $d = \sqrt{x^2 + y^2}$ the distance from the centre of the domain, using a Gaussian distribution such as

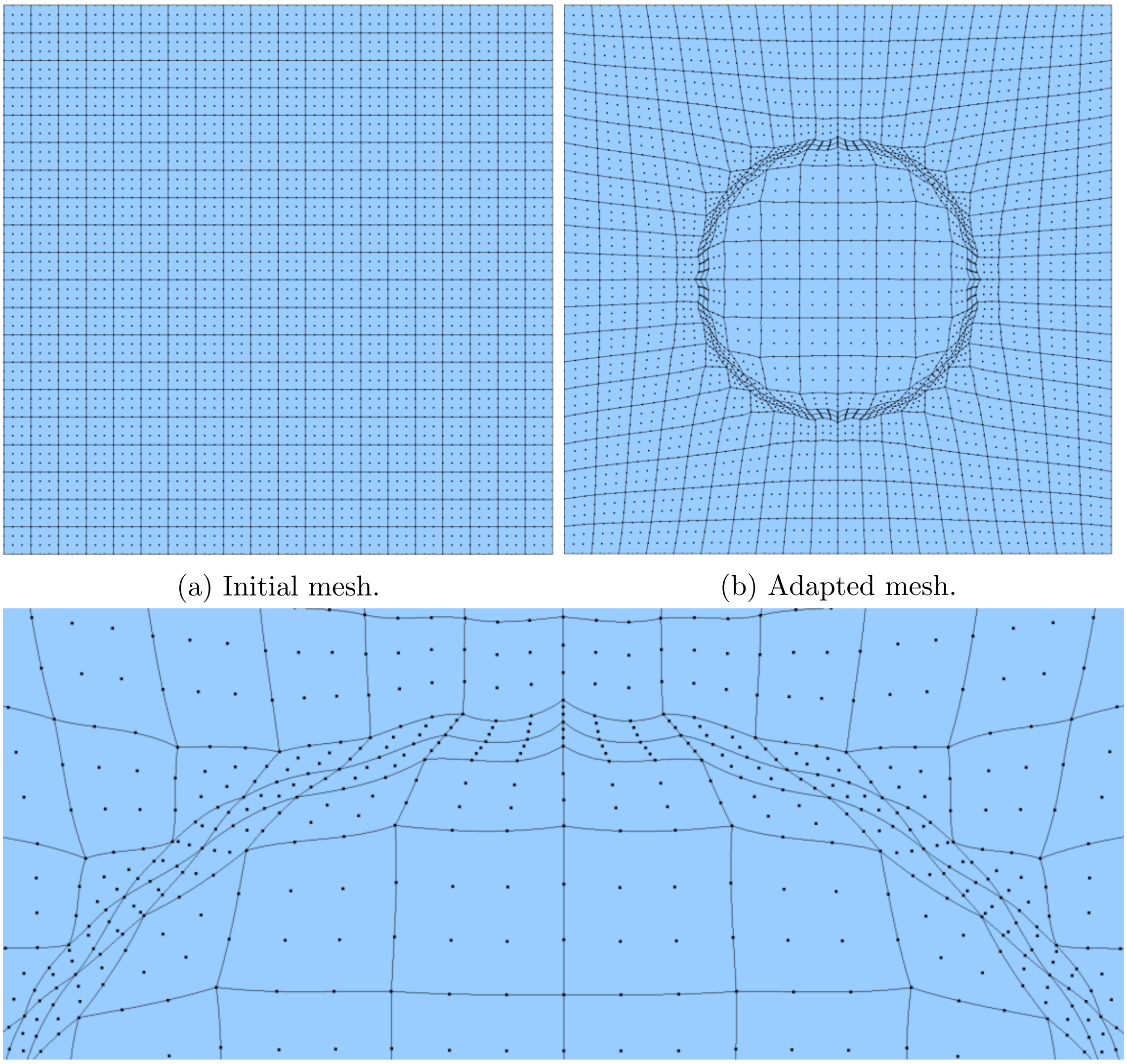
$$r(d) = 1 - \frac{A}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{5.9}$$

with the mean $\mu = 0.5$, the standard deviation $\sigma = 0.05$ and $A = 0.9\sqrt{2\pi\sigma^2}$ such that $\min_d r(d) = 0.1$.

Results are shown for a quad mesh in Fig. 5.2. The adapted mesh in Fig. 5.2b shows important refinement in the unit diameter circumference area. Coarsening is also observed everywhere else with bigger elements noted inside the circle. Such coarsening is to be expected as nodes are moved towards the unit diameter circumference and therefore stretch elements in the rest of the domain. It can also be observed from Fig. 5.2c that adaptation is indeed anisotropic: elements are shrunk in the radial direction only, keeping the size in the angular direction constant. However, elements are distorted; this is discussed below.

Another example is shown in Fig. 5.3. This domain corresponds to the truncated upper-right quadrant of the previous example, meshed this time by triangulation. The method behaves equally well for a less uniformly distributed triangular mesh as shown in Fig. 5.3b. The reader should note that this is a smaller domain, not a zoom in on a bigger domain, therefore demonstrating the *CAD sliding* capabilities in *NekMesh* for nodes on the boundaries.

In both examples, highly curved elements can be observed in the vicinity of the unit diameter region, which is indeed the expected behaviour. At the present time, a single metric tensor is used per element, computed at the barycentre of the linear element. This results in linear target elements. However, because high-order nodes are also optimised, elements are deformed by the algorithm in ways not intended by the metric tensor. To avoid this kind of behaviour, all subsequent r-adaptation is applied onto the linear mesh and the adapted mesh is then made high-order. This also allows a considerable speed up in compute time.



(a) Initial mesh.

(b) Adapted mesh.

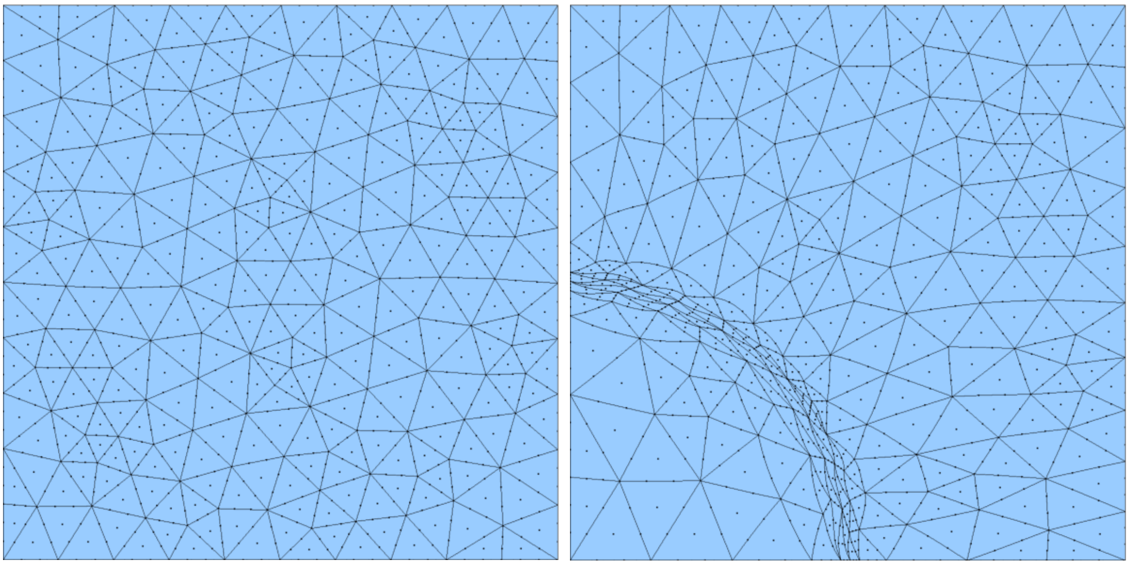
(c) Close-up view on the adapted mesh.

Figure 5.2: Anisotropic adaptation of a uniform quadrilateral mesh.

5.3 Error-driven adaptation

This section explores the use of an *a posteriori* error estimator to drive the r-adaptation procedure. Unlike traditional h-adaptation, r-adaptation has one major limitation: the number of elements, and therefore DOF, is constant. Regions of the mesh can only be refined by coarsening others. Additionally the mesh connectivities are preserved. Decreasing or increasing the size of elements directly translates into the deformation of neighbouring elements. These considerations must be taken into account when driving r-adaptation using an error indicator.

This section, unlike the rest of the thesis, makes use of *Aghora* [65], a high-order DG solver, for the inviscid compressible flow simulations. The details of the formulation and implementation of this solver can be found in the cited article and



(a) Initial mesh. (b) Adapted mesh.

Figure 5.3: Anisotropic adaptation of a triangular mesh.

are beyond the scope of this thesis.

5.3.1 Error indicator

To identify the target element size, the definition of an error estimator, providing an indication of the distribution of the local error over the spatial discretisation, is required. To this purpose, a local measure in an element K of the L^2-norm of the discretisation error of momentum is ideally employed:

$$\epsilon_K := \left(\int_K \left((\rho \mathbf{v})_{h,p} - (\rho \mathbf{v})_{ex} \right)^2 \mathrm{d}x \right)^{\frac{1}{2}}, \tag{5.10}$$

with the global error being defined as

$$\epsilon_\Omega := \left(\int_\Omega \left((\rho \mathbf{v})_{h,p} - (\rho \mathbf{v})_{ex} \right)^2 \mathrm{d}x \right)^{\frac{1}{2}}, \tag{5.11}$$

where ex denotes the exact solution.

The exact solution is in general unknown. The VMS[1] error estimator, previously used for p-adaptation [55, 56], is considered here. This error indicator estimates ϵ_K in Eq. (5.10) at order $p-1$ by considering the error between the numerical solution

[1] VMS: variational multiscale.

$(\rho\mathbf{v})_{h,p}$ and its projection on a reduced order space $(\rho\mathbf{v})_{h,p-1}$:

$$e_K := \left(\int_K \left((\rho\mathbf{v})_{h,p-1} - (\rho\mathbf{v})_{h,p}\right)^2 \mathrm{d}x\right)^{\frac{1}{2}}. \tag{5.12}$$

Despite its simplicity and conservative nature, this indicator has shown good performances for guiding p-adaptation procedures [55, 56]. It has been retained for this work because of the smoothness of its field, providing better conditioning for the r-adaptation process. An example is shown in Fig. 5.4 for an inviscid compressible flow over a Gaussian bump at $M = 0.5$, $p = 2$.

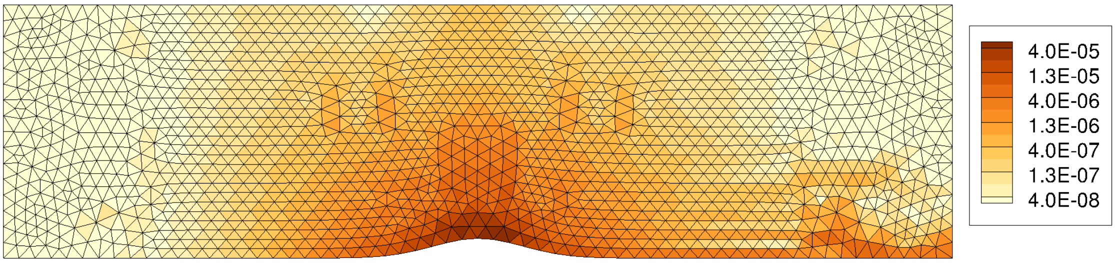


Figure 5.4: VMS error estimates for a Gaussian bump.

5.3.2 Target element size

From this error indicator, it is now necessary to determine a target element size to inject into the r-adaptation module of Sec. 5.2. Because the VMS error indicator is isotropic, only isotropic r-adaptation is considered.

It is known that for a high-order DG formulation, the discretisation error decays exponentially at a rate of $p + 1$ under uniform refinement such that

$$\epsilon = Ch^{p+1}, \tag{5.13}$$

where C is a constant that depends on the mesh and the regularity of the solution, and h the local element size. The theoretical decay rate can be obtained only under uniform refinement and is valid for $h \to 0$. Here it is loosely assumed that a similar relation is also valid for the local error and for the initial spatial discretisation. By making this assumption, an expression for the target element size can be obtained. First the current error e_K and target error e'_K of a single element are defined as

$$e_K = Ch_K^{p+1}, \qquad e'_K = C(h'_K)^{p+1}, \tag{5.14}$$

which give a measure of the target element size h'_K with respect to the initial element size h_K:

$$h'_K = h_K \left(\frac{e'_K}{e_K} \right)^{\frac{1}{p+1}}. \tag{5.15}$$

The target element size now only requires the definition of a target error e'_K.

5.3.3 Target error

Three different approaches are considered here to determine the target error $e' = e'_K$ for all elements K. They are later tested and analysed in Sec. 5.3.5.

Method 1: Constant L_2-norm

In the first approach, it is assumed that the total L_2-norm is unchanged:

$$\int_\Omega \left((\rho \mathbf{v})_{h,p} - (\rho \mathbf{v})_{h,p-1} \right)^2 \mathrm{d}x = \sum_K (e'_K)^2 = N(e)'^2, \tag{5.16}$$

where N is the number of elements. Thus,

$$e' = \sqrt{\frac{\sum_K e_K^2}{N}}. \tag{5.17}$$

Method 2: Maximum size reduction

In the second approach, a maximum size reduction is defined *a priori* for the element with the largest error. This allows for better control of the change in size imposed to the optimiser. This also limits elements with very large error to shrink, and therefore distort the mesh, too much. With q the minimum value of $\frac{h'_K}{h_K}$, assuming $p \geq 0$, the following condition is obtained:

$$\min_K \frac{h'_K}{h_K} = \min_K \left(\frac{e'}{e_K} \right)^{\frac{1}{p+1}} = \left(\frac{e'}{e_{\max}} \right)^{\frac{1}{p+1}} = q, \tag{5.18}$$

where q is a user parameter and chosen to be 2 in this work. Thus,

$$e' = e_{\max} q^{p+1}. \tag{5.19}$$

Method 3: Constant domain size

In the third and last approach, the fixed total domain size constitutes the starting point. In two dimensions, this is expressed as

$$\sum_K {h'_K}^2 = \sum_K h_K^2. \tag{5.20}$$

After adaptation, the surface area covered by the r-adapted mesh will be the same as that covered by the initial mesh. From Eq. (5.20), the value of e' can then be derived by substituting the expression for $h'_K(e')$.

5.3.4 Limit deformation

Large variations in target element size can be obtained from Eq. (5.15). In particular, very low values of the error indicator in areas of free stream conditions, for example, yield very large target sizes. This can practically hinder the convergence of the optimisation procedure, due to the fixed mesh connectivities of r-adaptation. To limit this deformation, two constraints are considered here:

- Limit the maximum growth rate $r_{\max} = h'_K/h_K$;
- Limit the maximum absolute size $h'_{\max} = \max h'_K$.

In the *constant L_2-norm* approach of Eq. (5.17) and the *maximum size reduction* approach of Eq. (5.19), these constraints are applied to h'_K. In the *constant domain size* approach of Eq. (5.20), however, the value of e' must be updated by solving

$$\sum_K h'_K(e')^2 = \sum_K {h_K}^2. \tag{5.21}$$

5.3.5 Application

The test case of Fig. 5.4 is used here to illustrate the different approaches and limit constraints. It consists of an inviscid compressible flow over a Gaussian bump at $M = 0.5$, $p = 2$. There are no discontinuities and the flow is reversible. The L^2-norm of the error is calculated with respect to a very fine reference solution. All CPU times are reported in seconds on a single processor.

An initial solution is calculated in *Aghora* on a uniform unstructured mesh, shown in Fig. 5.4. From this solution, the VMS error indicator is exported to *NekMesh* where the mesh undergoes r-adaptation following the conversion of e_K into e' and, subsequently, h'_K using one of the three approaches described above with possibly a limit constraint. The mesh then returns to *Aghora* where the cycle is repeated. The process goes on until the L^2-norm of the error reaches a minimum for this given number of DOF and mesh connectivities.

First, it is interesting to note that, to the naked eye, the final mesh looks very similar for all cases. Fig. 5.5 shows a typical final mesh after convergence is reached in the r-adaptation iterations. Note that this mesh has a lower element density than the initial mesh shown in Fig. 5.4.

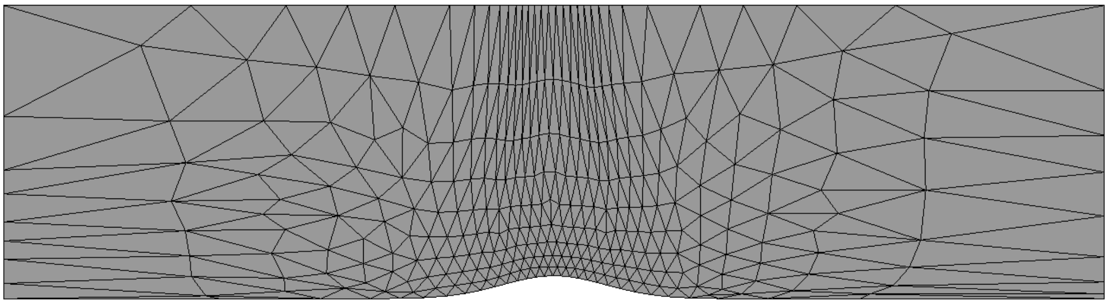


Figure 5.5: Example of a final r-adapted mesh.

Figs. 5.6, 5.7 and 5.8 first compare the different limit constraints for each of the three approaches. Only in the *maximum size reduction* approach, the limit constraint makes a difference in the convergence history. In this approach, the *maximum growth rate* constraint seems to aggressively limit the deformation of the mesh and convergence is reached slower. The same final value of L^2-norm of the error is however still reached.

The right hand side figures each correspond to the corresponding CPU[2] time to solve the problem on the corresponding meshes, thus giving insight into the quality of the meshes. In all cases, r-adapted meshes take longer to solve. The *maximum growth rate* constraint, however, generates meshes that are faster to solve for a given L^2-norm of the error. This seems consistent with the previous observation, proving that this constraint tends to better reduce the deformation of the mesh.

Figs. 5.9 and 5.10 then compare the different approaches for each limit constraint. The *constant L_2-norm* approach seems to converge the fastest at first while the *constant domain size* approach gives the lowest final L^2-norm of the error. In terms of CPU time, it is the *maximum size reduction* approach that outperforms the other two for a given L^2-norm of the error, with the *constant L_2-norm* approach consistently the slowest. From this, one can conclude that the *constant L_2-norm* approach is the most aggressive of the three approaches and the *constant domain size* approach a decent compromise between deformation and speed of convergence.

[2] CPU: central processing unit.

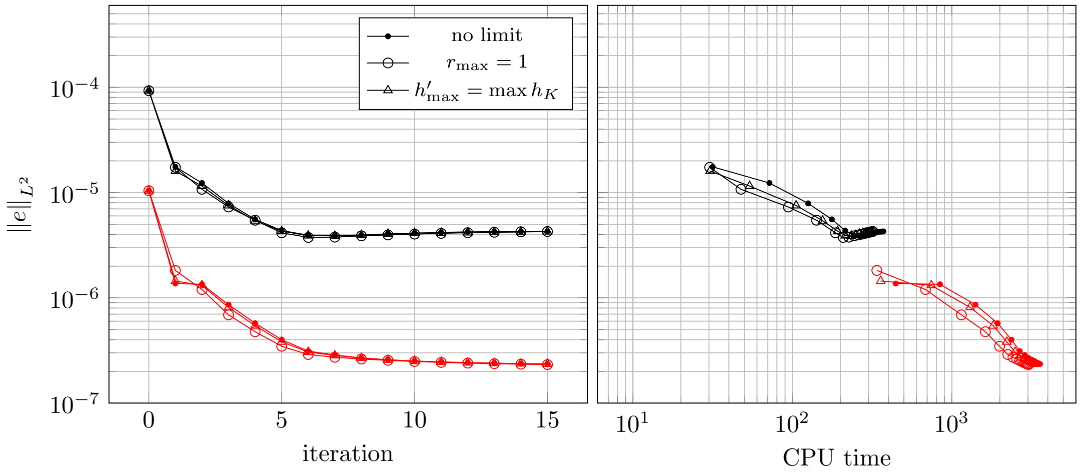


Figure 5.6: Convergence history with the *constant L_2-norm* approach for two different mesh densities: coarse in black, fine in red.

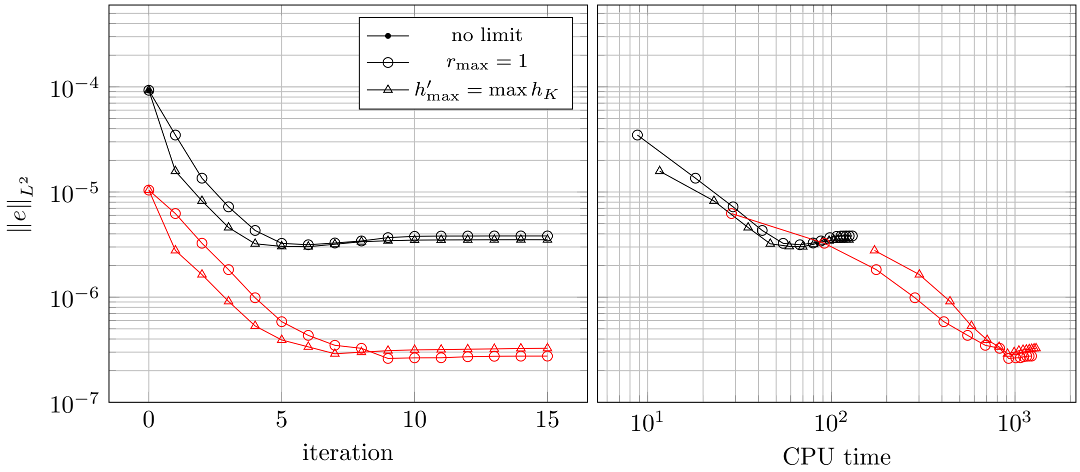


Figure 5.7: Convergence history with the *maximum size reduction* approach for two different mesh densities: coarse in black, fine in red.

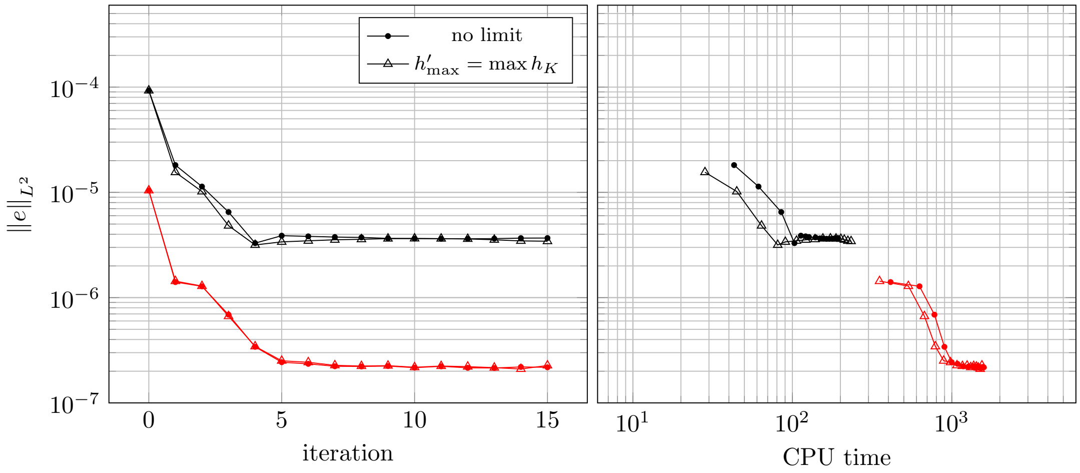


Figure 5.8: Convergence history with the *constant domain size* approach for two different mesh densities: coarse in black, fine in red.

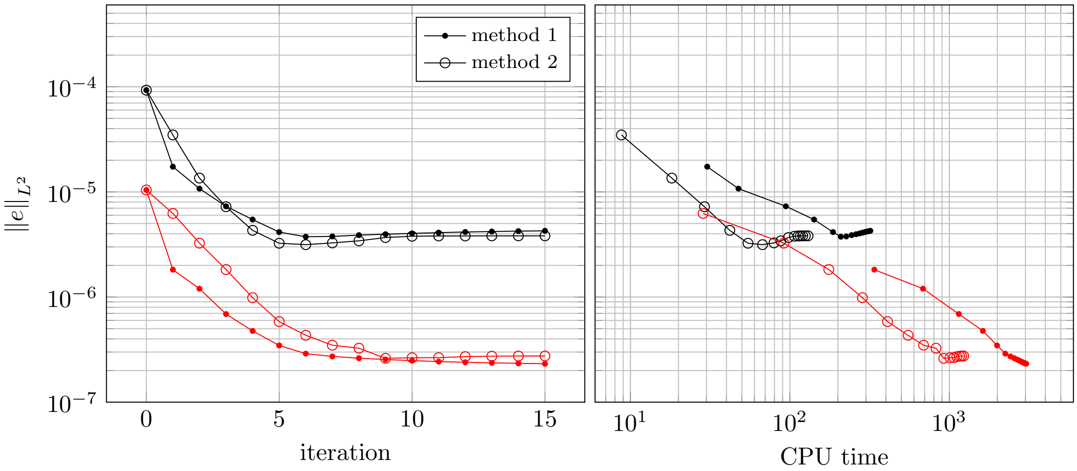


Figure 5.9: Convergence history with the *maximum growth rate* constraint for two different mesh densities: coarse in black, fine in red.

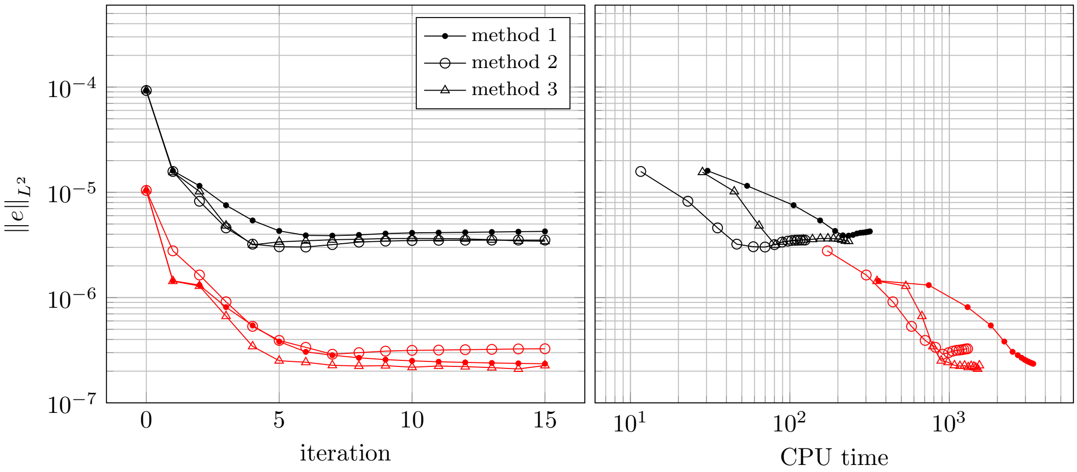


Figure 5.10: Convergence history with the *maximum absolute size* constraint for two different mesh densities: coarse in black, fine in red.

In all cases, mesh convergence is reached within typically 10 iterations, sometimes faster. Importantly, the quality of the mesh does not degrade at any point and the error remains constant after reaching convergence.

Chapter 6

rp-adaptation

This chapter proposes to use r-adaptation, as derived in the previous chapter, with a p-adaptation strategy already present in *Nektar++*. The advantageous properties of these two strategies are combined in the same simulation to maximise their effect in increasing the resolution of shocks and decreasing the computational cost of compressible flow simulations. More specifically, r-adaptation will be responsible for the resolution of shocks whereas p-adaptation will resolve smooth flow regions. p-adaptation is used in the smooth flow regions because of its higher convergence rates [15, 40, 24]. This advantage, however, does not apply to regions of flow discontinuities, where h-type adaptation is preferred. In this work, r-adaptation is used at shocks to achieve h-like refinement.

These two adaptation strategies are applied one after the other in a sequential approach proposed in Sec. 6.3 while Sec. 6.1 and Sec. 6.2 describe the specific uses of r- and p-adaptation in this workflow. Two numerical examples are then presented and discussed in Sec. 6.4.

6.1 Improving shock resolution via *r*-adaptation

Although in principle the target element $\mathbf{\Omega}_T^e$ can take any shape and size, a practical approach was adopted in this chapter that aims to avoid too large deformations. The rationale for this is that the definition of a target element $\mathbf{\Omega}_T^e$ that is very different from the ideal element $\mathbf{\Omega}_I^e$ — i.e. the initial linear element before r-adaptation — introduces extra energy in the system that the optimiser has to minimise and thus slows down the process. For this reason, a target element $\mathbf{\Omega}_T^e$ is defined with respect to the ideal element $\mathbf{\Omega}_I^e$. In this chapter, elements are *shrunk* isotropically where additional resolution is required, i.e. in the shock regions. As shown in Sec. 5.2, this is accomplished by scaling the Jacobian of the mapping $\boldsymbol{J} = \nabla\boldsymbol{\phi}$ by a factor r:

$$\boldsymbol{J}_T = (r\boldsymbol{I})\boldsymbol{J}_I. \tag{6.1}$$

6.2 Reducing overall cost via p-adaptation

To enhance resolution in regions of smooth flow through local p-adaptation, the following, fairly straightforward, procedure [24, 51] is used. The local resolution is improved by increasing the polynomial order within the elements where the local error is estimated to be high and vice versa. This procedure is summarised in Alg. 4 where e denotes an individual element, s_e and p_e are its associated error indicator and polynomial order, ε_u and ε_l are the upper and lower error thresholds, and $p_{\max}$ and $p_{\min}$ are the maximum and minimum polynomial orders allowed. In this work, the formulation of sensor s_e in Eq. (2.24) is used as the indicator of the discretisation error.

```
repeat
    Calculate the steady-state solution ;
    forall $e$ do
        Calculate $s_e$ ;
        if $s_e > \varepsilon_u$ and $p_e < p_{\max}$ then
            Increment $p_e$ ;
        end
        else if $s_e < \varepsilon_l$ and $p_e > p_{\min}$ then
            Decrement $p_e$ ;
        end
        else
            Maintain $p_e$ ;
        end
    end
until no $p_e$ is modified;
```

Algorithm 4: Employed p-adaptation procedure.

6.3 Workflow

In the proposed rp-adaptation workflow, these adaptive techniques will be alternatively applied in a sequence of three steps.

First an initial high-order mesh is generated for the domain. The requirements of r-adaptation and the need for DOF to be moved around when deforming the mesh are anticipated. For this reason, a relatively coarse mesh is generated, but with enough resolution to allow for node movement. The solver is then run on this initial mesh and a flow solution obtained which represents the base solution. During this step, appropriate parameter values must be determined for the artificial viscosity. As is common practice in codes based on artificial viscosity, *Nektar++* default values are first used and then tuned for the specific problem. The artificial viscosity parameters (s_κ and κ) are adjusted to ensure that artificial viscosity is only triggered in the direct vicinity of shock waves, and the level of artificial viscosity (μ_0) adjusted so that the shock is stable but not overly dissipative.

From this base solution, r-adaptation is applied to the mesh. The list of elements where artificial viscosity was added during the initial simulation is first extracted. If the run was set up properly, these elements only represent the regions where a shock is present. The barycentre of these elements is extracted and an isotropic shrinking factor r is assigned to them (see Sec. 6.1). For all the other elements, the barycentres are also extracted and assigned a factor $r = 1$. In practical terms, elements in the shock regions are forced to shrink and pull mesh nodes from all other parts of the mesh. This field of r factors is then supplied to the variational r-adaptation code which is then run on the linear mesh. The variational framework optimises the mesh so that each element is as close as possible to its target size, effectively moving nodes from areas of $r = 1$ to areas of $r < 1$. Importantly r-adaptation is run on the linear mesh before making the r-adapted mesh high-order again. This significantly speeds up the optimisation procedure and improves the validity of the final mesh. The solver is then run on the r-adapted high-order mesh and a new solution is obtained with enhanced shock resolution. This procedure can optionally be repeated based on the new solution.

From this solution on the r-adapted mesh, p-adaptation can be run as described in Sec. 6.2. At the end of each cycle, a sensor value is computed for each element and the local polynomial order of that element is decreased, kept the same or increased based on the value of the sensor. In this work, *Nektar++* default values are used for the p-adaptation parameters: $\varepsilon_u = -6$, $p_{\max} = 6$, $\varepsilon_l = -8$ and $p_{\min} = 2$. The simulation then proceeds onto a new cycle and the process is repeated until a steady solution is obtained and the local polynomial orders do not vary.

The user can also choose to restrict the polynomial order of elements within the shock regions. These are the zones that have been previously identified in the r-adaptation procedure. The local polynomial order of the elements in these regions is then set to a user-defined value, which should be typically low ($p < 3$). The proposed rp-adaptation workflow is summarised in Alg. 5.

```
Generate an initial high-order mesh ;
Calculate the steady-state solution ;
repeat
    Extract shock areas based on sensor values ;
    Apply $r$-adaptation in shock areas to linear mesh and re-project to
      high-order ;
    Calculate the steady-state solution ;
until shocks are well captured;
Apply $p$-adaptation as described in Alg. 4 ;
Calculate the final solution ;
```

Algorithm 5: Proposed *rp*-adaptation workflow.

6.4 Application

In this section, two different test cases are used to demonstrate the *rp*-adaptation workflow: a NACA 0012 aerofoil in transonic regime and a supersonic intake. Different difficulties arise for each of these test cases as will be discussed below. Most importantly, slightly different approaches are taken when it comes to r-adaptation and p-adaptation.

For the first test, only two simple shocks are present on the upper and lower surfaces of the aerofoil. It is easy for the variational framework to pull mesh nodes from the smooth regions and a single round of r-adaptation is therefore run. For the second test, a complex diamond-like pattern of oblique shocks is expected due to reflections inside the internal configuration of the intake. The number of nodes available for deforming the mesh is limited, which places additional stress on the variational optimisation process. A two-step approach to r-adaptation is therefore used where each step is run with a milder shrinking factor in order to retain good mesh quality.

In terms of p-adaptation, the first test case is used as a benchmark for this order restriction strategy. Three different approaches to order restriction are studied. In the first, no restriction is applied and the local polynomial order of elements in shock areas is left free to increase. In the second, the local polynomial order of these elements is kept constant, i.e. the order of the initial simulation. Finally, in the third case the local polynomial order of these elements is immediately decreased to the minimum allowed in the run.

6.4.1 NACA 0012 profile

The new technology is first demonstrated on a canonical aeronautical test case: a transonic ($M = 0.8$) inviscid flow past a NACA 0012 aerofoil at 1.25° angle of attack.

This configuration produces two shocks [82]: a strong shock on the suction side and a weak shock on the pressure side at approximately 60% and 35% of the chord respectively. This provides a relatively easy test case to showcase the technology where the shocks are quasi-vertical. The main difficulty lies in the relative weakness of the shock on the pressure side and in capturing it appropriately.

The domain used has external boundaries at a distance of $40c$ from the aerofoil, where c denotes the chord length. The domain is discretised uniformly along the chord with an element size of $\sim 0.05c$ on the aerofoil boundary and a smooth progression is used towards an element size of $\sim 10c$ on the outer boundary. The automatic sizing of elements in the field is determined through an octree system [79]. The mesh is curvilinear of order $p = 4$ and it is optimised using the variational framework described in Sec. 5.1. The worst scaled Jacobian for this mesh is found to be $J_{s,\min} = 0.770$ after optimisation. Fig. 6.1a shows what the mesh looks like in the near field. The starting mesh is relatively coarse but it is run through the compressible flow solver described in Sec. 2.2 at uniform $p = 4$ order. This order $p = 4$ is kept throughout the procedure until p-adaptation is finally applied. The reader should note the importance of having sufficient resolution (either through h or p) in the initial mesh in order to distinguish shocks, i.e. actual discontinuities, from smooth high-gradient regions when looking at high discontinuity sensor values.

The solver is first run on the initial mesh to obtain a base solution. Slip wall BC are imposed on the surface of the profile and far-field BC at the external boundaries of the domain. The HLLC[1] Riemann solver [76] is used. For the artificial viscosity, the solver parameters are tuned to $s_\kappa = -1.2$, $\kappa = 0.7$ and $\mu_0 = 1.0$. Fig. 6.1e shows that large values of the sensor are obtained in both shock areas but also near the leading and trailing edges. However, Fig. 6.1g shows that artificial viscosity only triggers in the vicinity of the two shocks, proving adequate tuning of the artificial viscosity parameters. As a result, the simulation is stable and steady state is reached. The flow solution in Fig. 6.1c displays very thick shocks as expected on this relatively coarse mesh. Oscillations are also observed further in Fig. 6.2 in the field past the strong shock caused by the under-resolution of the shock and the generation of entropy.

r-adaptation

From the base solution, the workflow explained in Sec. 6.3 is employed. The shock regions are first extracted: these correspond to the elements of non-null artificial viscosity in Fig. 6.1g. To these regions, a shrinking factor of 0.1 is assigned and

[1] HLLC: Harten-Lax-van Leer-Contact.

the r-adaptation procedure is run. A new mesh is obtained which, for quality considerations, is re-optimised before simulation. The worst scaled Jacobian for this mesh is found to be $J_{s,\min} = 0.648$ after optimisation. This is mostly due to the coarsening of element at the leading edge. The new mesh shown in Fig. 6.1b shows refinement in the shock areas and consequently a slight coarsening outside of those zones. Shrinking is also observed, to a smaller extent, in the vertical direction due to the isotropy of the r-adaptation approach. However, the resulting mesh is clearly anisotropic and aligned to the presence of the shock.

The old solution is now interpolated onto the r-adapted mesh and the simulation run again. In order to avoid any instability of the solver due to the interpolation of the under-resolved shock onto the new mesh, the solver is first run over a few hundred time steps with a decreased step size. The simulation is then run, using the exact same artificial viscosity parameters, until steadiness is achieved. The flow solution in Fig. 6.1d shows better resolution of both shocks as seen by the sharpness of the shocks. Reduced oscillations are also observed in the wake of the strong shock. Figs. 6.1f & 6.1h finally show that discontinuity, as per the sensor, is now observed in a narrower area and that the artificial viscosity reaches lower values.

The improvement of the resolution of the shock can be better seen in 2D plots. Fig 6.2 shows the Mach number on the surface of the profile. Because elements are so large in the initial mesh, the solution shows strong oscillatory behaviour, known as the Gibbs phenomenon. After r-adaptation, elements in the region of the shock are much smaller and, although oscillations are still present, they have both a smaller amplitude and a narrower range. This confirms the qualitative observation of the increased sharpness of the shock seen in Fig. 6.1d. Because there is less mesh movement at the weak shock, the reduction in the Gibbs phenomenon is also smaller.

To retain accurate boundary representation, it is important that the r-adaptation code has access to a CAD system. In this instance, *NekMesh* was implemented to use the *Open Cascade* [58] kernel as its CAD engine under a small wrapper layer. This allows *NekMesh* to query the geometry and ensure that all nodes remain on the boundaries at all time, allowing *CAD sliding*. This capability is shown in Fig. 6.3 where nodes remain on the aerofoil surface throughout the r-adaptation process. Fig. 6.3 also shows that the optimiser is able to move nodes across large distances, as seen through the row of coloured elements before (Fig. 6.3a) and after (Fig. 6.3b) r-adaptation.

Fig. 6.3 further demonstrates the strong anisotropic nature of the r-adaptation procedure, although the target element is shrunk isotropically. Shrinking occurs predominantly in the horizontal direction while elements mostly retain their initial

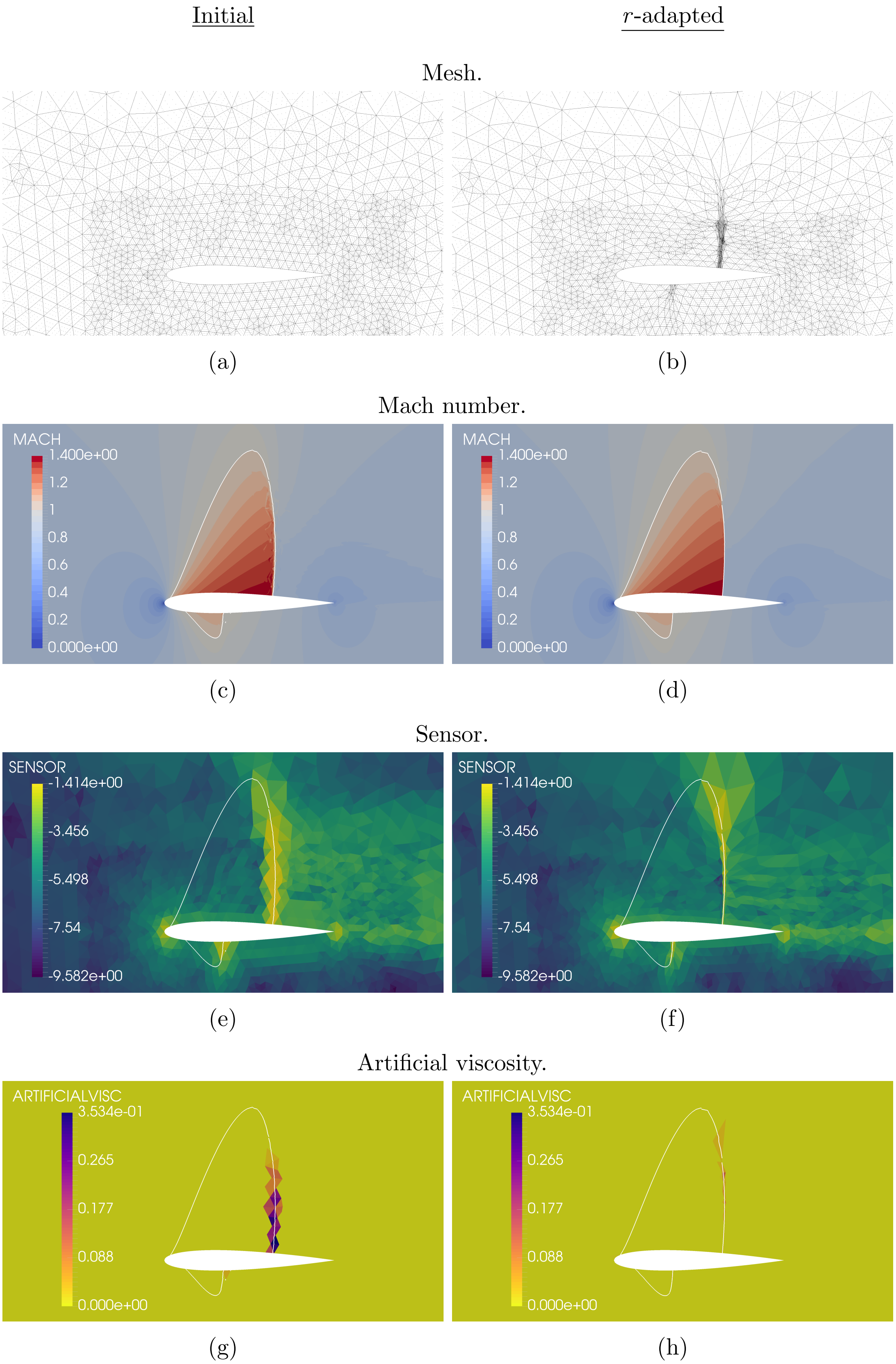


Figure 6.1: Mesh and fields for the NACA 0012 profile before and after the round of r-adaptation: white line denoting $M = 1$.

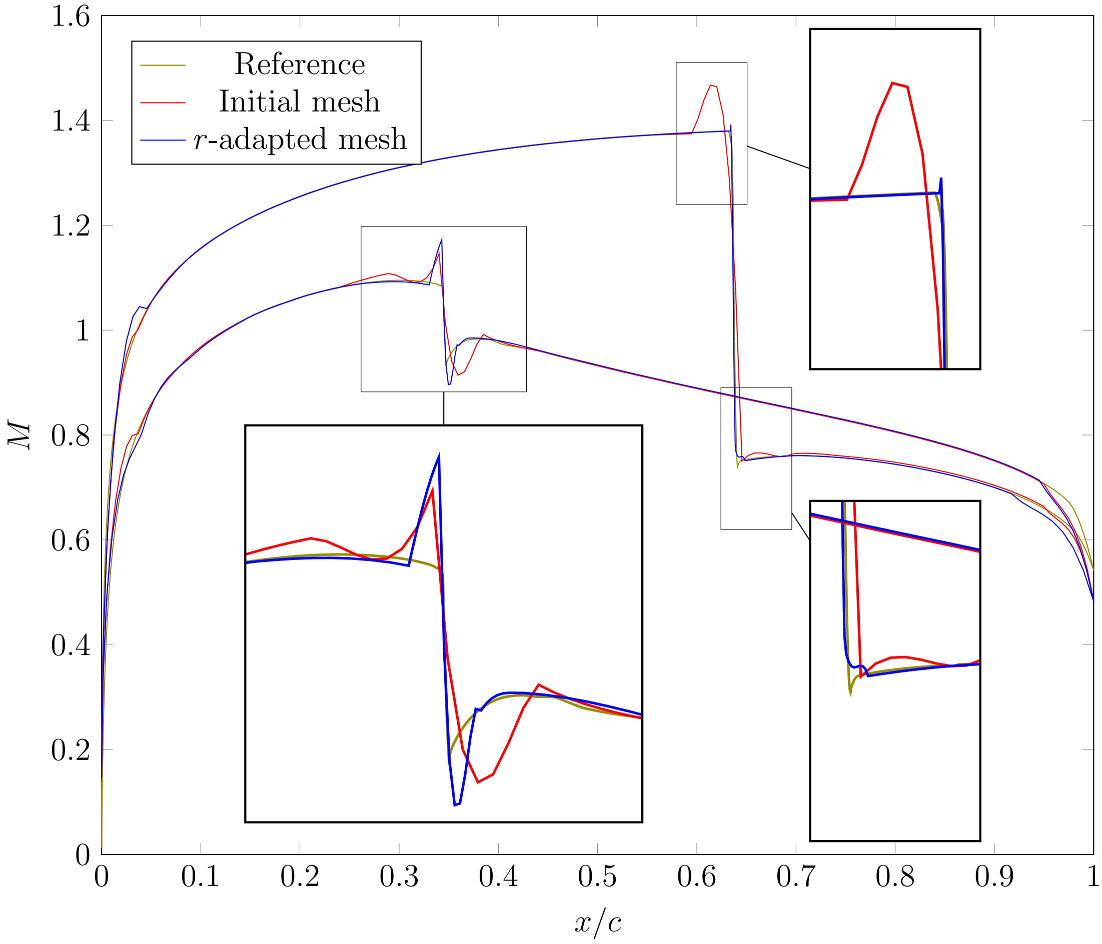
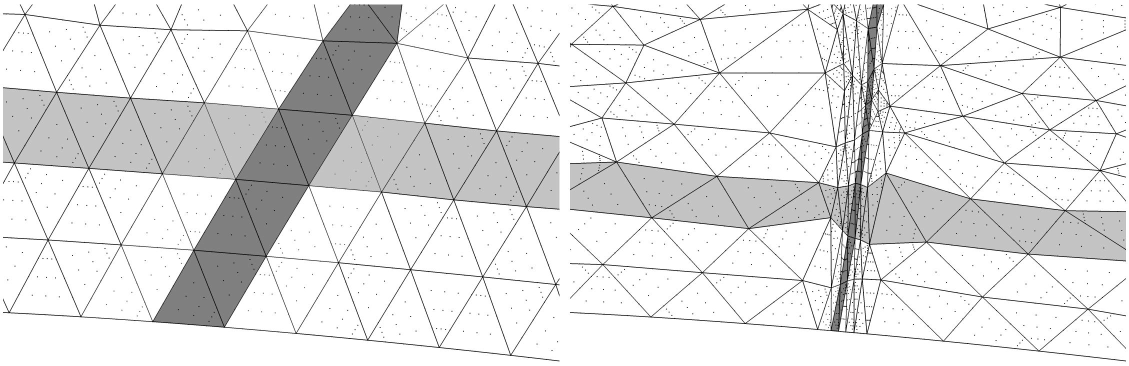


Figure 6.2: Plot of the Mach number before and after the round of r-adaptation.



(a) Initial mesh.

(b) r-adapted mesh.

Figure 6.3: Close-up view on boundary elements during *CAD sliding*: elements coloured identically.

height, both inside and outside the shock region. This is easily explained by the balance in energy in the system. The optimiser finds a minimum in deformation energy when elements are only mildly shrunk vertically. Indeed, while elements in the shock are trying to pull vertices from all directions, incl. vertically, elements outside the shock are trying to keep their size. This tension reaches equilibrium in the mesh shown in Fig. 6.3b. This tension does not exist, however, in the horizontal direction, in which case vertices are allowed to freely shrink elements to best match the size of their target element.

p-adaptation

After better resolving the shocks, local p-adaptation can now be applied for the smooth field. For this test case, three scenarios are compared. In the first scenario, local p-adaptation is applied without any restriction (see Figs. 6.5a) while, in the other two scenarios, the local polynomial order is restricted inside the shock areas. In the second scenario, the local polynomial order of the uniform $p = 4$ order simulation of Sec. 6.4.1 (see Fig. 6.5c) is preserved. In the third and last scenario, the local polynomial order inside the shock areas is decreased to the lowest user-allowed order (see Fig. 6.5e).

For these tests, the field obtained at $p = 4$ in Sec. 6.4.1 is used as initial conditions and values of $p_{\text{min}} = 2$ and $p_{\text{max}} = 6$ are used. The sensor is based on the density ρ field and the solver default values of lower and upper sensor tolerances are respectively 10^{-8} and 10^{-6}. Fig. 6.5 shows the results. Figs. 6.5a, 6.5c, 6.5e show a final map of the local number of modes ($= p + 1$) after a steady solution is reached and, by extension, when the local polynomial order remains constant throughout p-adaptation steps. The number of DOF for each simulation is shown in Table 6.1. All scenarios produce fewer DOF than the simulation at uniform p, thanks to local p-coarsening in low-error regions. As expected, the unrestricted p-adaptation scenario increases the local polynomial orders of elements in the shock thickness to the maximum user-allowed value. This leads to a higher global number of DOF than the other two scenarios. Then follows the second scenario while the last scenario has the smallest number of DOF. Each of these DOF counts also translates into similar increases or decreases in compute times.

These solutions are compared to a reference solution computed on a very fine mesh. To evaluate the performance of each mesh and p-adaptation scenario, the Mach number distribution on the surface of the aerofoil is compared for each scenario. The L^2-norm of the error is used, defined as

$$\|e\|^2_{L^2(S)} = \int_S (M - M_{ref})^2 \, dS, \tag{6.2}$$

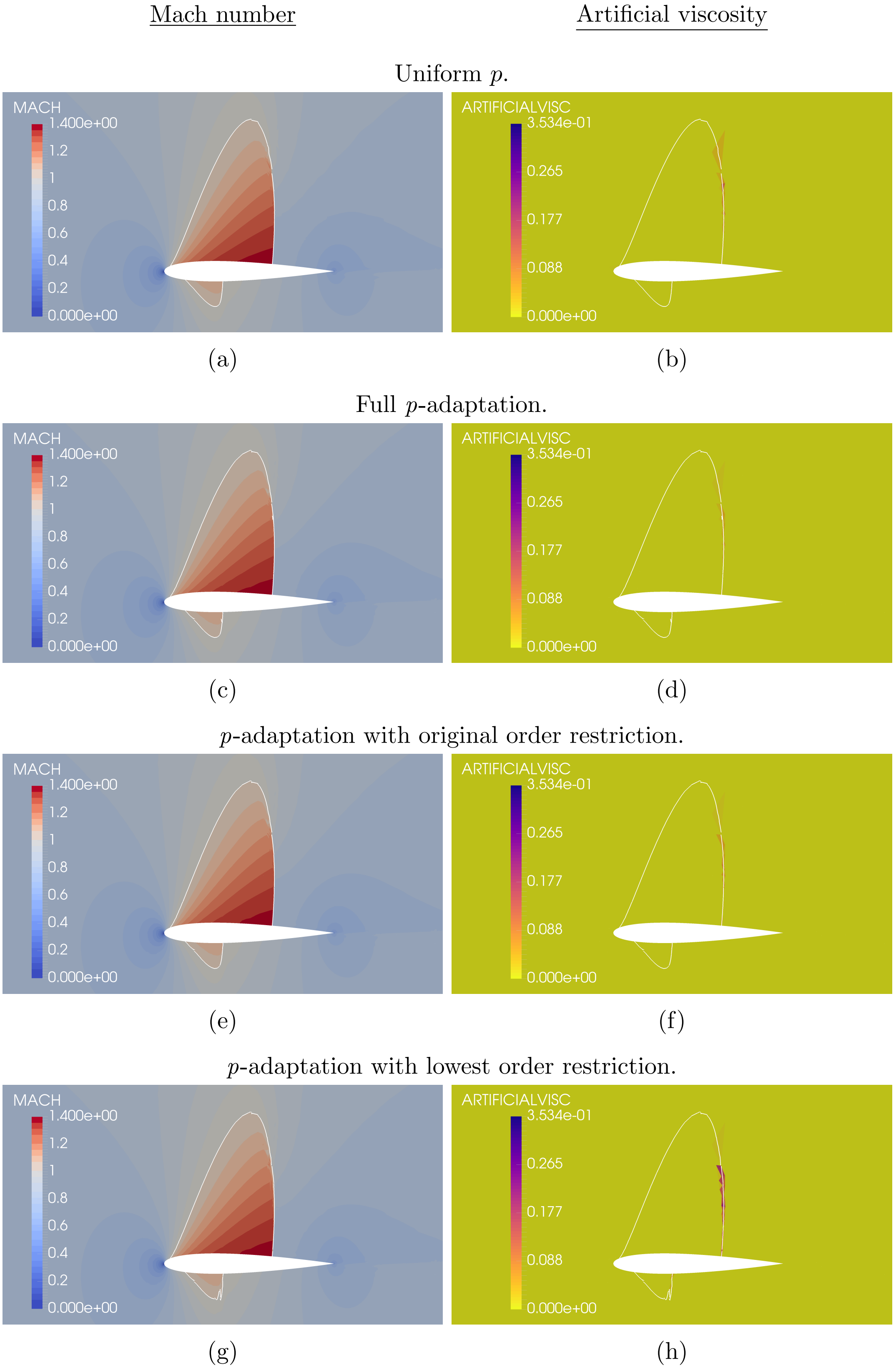
where M is the Mach number of the test solution, M_{ref} is the Mach number of the reference solution and S is the chord. Results are reported in Table 6.1. Note that CPU times per time step are reported as run on a 16-core machine, once convergence is reached. It is first noted that r-adaptation alone provides an important boost in terms of accuracy. Scenarios #1 and #2 both suffer a loss of accuracy due to the coarsening of the solution in large parts of the domain. This slight increase in the error, however, allows the number of DOF to be cut in half. Scenario #3, on the other hand, performs very poorly, with the error going even higher than on the initial mesh. Decreasing the polynomial order inside the shock — a rather small region — allows a few more DOF to be saved but at too great a cost.

Table 6.1: Number of DOF, error and CPU time per time step at convergence for the NACA 0012 profile.

Simulation	Number of DOF	$\|e\|^2_{L^2(S)}$ (10^{-4})			CPU time (ms)
		Pressure	Suction	Total	
Initial mesh	65 550	0.547	5.28	5.83	65
r-adapted mesh	65 550	0.751	1.36	2.11	118
Scenario #1	29 201	0.875	2.54	3.41	48
Scenario #2	29 117	0.919	1.82	2.74	56
Scenario #3	27 736	1.045	6.61	7.65	45

Fig. 6.4 shows a comparison of the Mach number and artificial viscosity fields for the uniform p simulation and the three test scenarios. Little difference is observed between scenarios #1 and #2. Scenario #3 on the other hand exhibits under-resolution of the shock, seen through its thicker profile. This is consistent with the local element size and the lack of DOF in the thickness of the shock at lower order. As a result, the last lower-order scenario exhibits some oscillations in the wake, due to the generated entropy in the shock area. It is also observed that lower-order scenarios induce more artificial viscosity. This phenomenon is consistent with the previous assessment of the lack of resolution of the shock. The discontinuity sensor detects a certain lack of resolution and therefore more artificial viscosity is added to the system.

Overall all scenarios expectedly exhibit similar distributions of local polynomial order in the smooth field regions in Fig. 6.5. When analysing the distribution of local polynomial orders, higher orders are observed in the area above the strong



Mach number

Artificial viscosity

Uniform p.

(a)

(b)

Full p-adaptation.

(c)

(d)

p-adaptation with original order restriction.

(e)

(f)

p-adaptation with lowest order restriction.

(g)

(h)

Figure 6.4: Mach number and artificial viscosity for the uniform p simulation and the three test scenarios of the NACA 0012 profile: white line denoting $M = 1$.

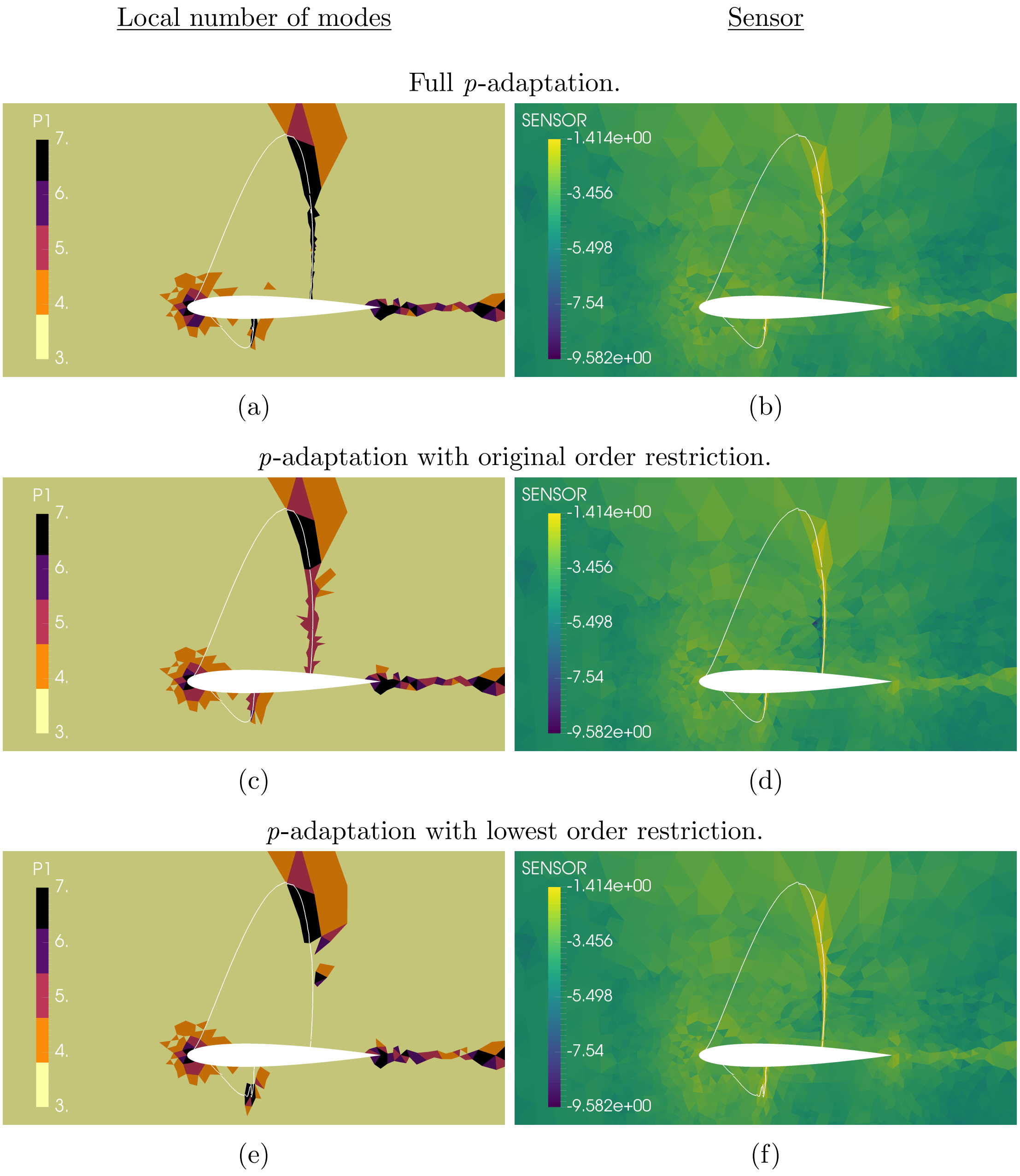
shock and below the weak shock, in all scenarios. These areas were not detected in Sec. 6.4.1 as part of the shock due to the then under-resolved and therefore too short shocks. Now that the shocks are better resolved, they reach further out and require additional resolution, in the form of higher polynomial order in this case. It is also observed, in the lower-order scenarios, that parasite higher-order zones are created. This is especially obvious around the weak shock in the third scenario. This is due, as noted above, to the thicker shock profile and therefore the need to add resolution around the shock. Fig. 6.5e is consistent with this explanation as a larger area of high sensor values is observed, extending beyond the shock areas determined in Sec. 6.4.1.

6.4.2 Supersonic intake

This section illustrates the new approach on a test case with a more complicated shock pattern. The test case is that of a supersonic intake at $M_\infty = 3.0$ first studied experimentally [2] and later numerically [34]. The intake consists of two straight ramps inclined with respect to the incoming free-stream flow at angles of 7° and 14° respectively. The first ramp creates an oblique shock which impinges on the horizontal cowl and in turn leads to a complex pattern of reflecting oblique shocks throughout the diffuser of the intake. The difficulty here is the presence of multiple shocks with different orientations in the very narrow regions of the diffuser.

The domain is discretised uniformly in the streamwise direction. An element size of $0.01L$ (L being the length of the intake) is used inside the intake and let it coarsen outside the intake up to an element size of $0.05L$ in the far-field. The mesh is curvilinear of order $p = 4$ and it is optimised in the throat. The worst scaled Jacobian for this mesh is found to be $J_{s,\min} = 0.925$ after optimisation. Fig. 6.6a shows what the mesh looks like inside the intake and in its immediate surrounding.

The solver is run at uniform order $p = 3$ on the initial mesh to obtain a base solution. This order $p = 3$ is kept throughout the procedure until p-adaptation is finally applied. Wall BC are imposed on the surfaces of the intake, outlet BC with a low enough pressure at the outlet until a fully supersonic field is obtained ($P_b = 0.9P_{\text{inf}}$) and far-field BC at the external boundaries of the domain. Roe's approximate Riemann solver [76] is used. For the artificial viscosity, the solver parameters are tuned to $s_\kappa = 0.0$, $\kappa = 0.0$ and $\mu_0 = 0.1$. Fig. 6.6e shows that large values of the sensor are obtained in all shocks and that moderate values are obtained everywhere after the first upstream shock. However, artificial viscosity only really triggers in the vicinity of the shocks as Fig. 6.6g shows, proving adequate tuning of the artificial viscosity parameters. A steady state solution is reached thanks to a



Local number of modes

Sensor

Full p-adaptation.

(a)

(b)

p-adaptation with original order restriction.

(c)

(d)

p-adaptation with lowest order restriction.

(e)

(f)

Figure 6.5: Local number of modes ($= p + 1$) and sensor for the three test scenarios of the NACA 0012 profile: white line denoting $M = 1$.

stable simulation. Just like for the NACA 0012 test case, the shocks demonstrate a thick profile, as can be seen in Fig. 6.6c due to the relatively coarse local mesh as well as some oscillations near the leading edge of the cowl.

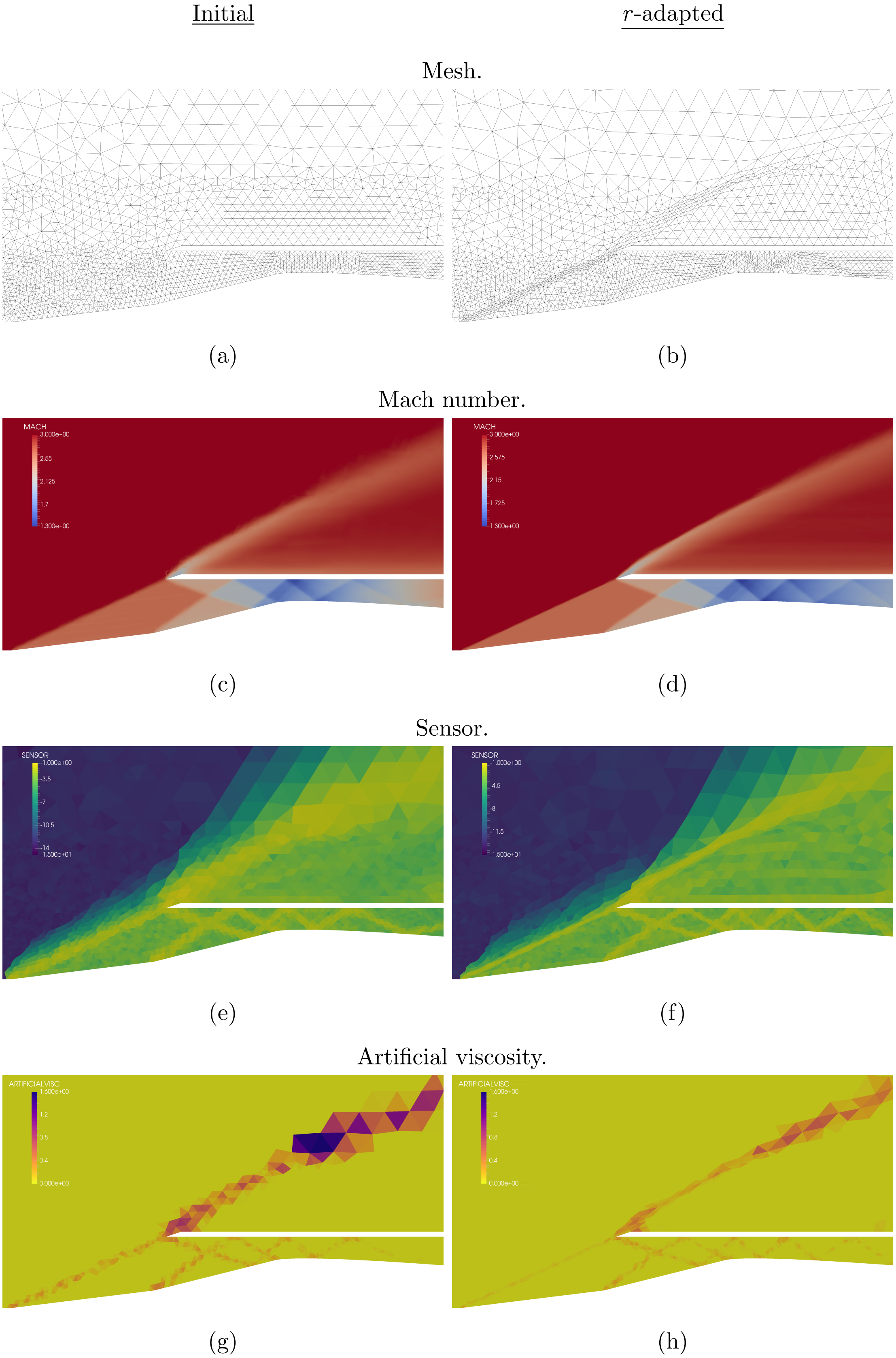
***rr*-adaptation**

The workflow laid out in Sec. 6.3 is again employed except that two rounds of r-adaptation are run, obtaining rr-adaptation. Each round uses a less aggressive shrinking factor of 0.5. Before each simulation, the mesh is again optimised for high-order mesh quality reasons. The worst scaled Jacobian for this mesh is found to be $J_{s,\min} = 0.985$ after optimisation, mostly thanks to the refinement of elements near the curvature at the throat. The new mesh after one round of r-adaptation is shown in Fig. 6.6b. Refinement is observed in all areas of interest although stronger in the area of the first upstream shock. Indeed, elements in the first shock are able to pull DOF from the freestream areas whereas elements inside the intake are interacting with each other. Refinement is nonetheless obtained in all shock areas and anisotropy naturally appears such that elements are shrunk in mostly the shock normal direction. The r-adaptation strategy works by pulling nodes together. Because the shrinking areas are long and narrow, nodes are naturally moved normally rather than tangentially to the underlying shock, without the need for the optimiser to be aware of the shock structures.

The solver is now run on the new r-adapted mesh using the same solver parameters. A stable flow solution is obtained and shown in Fig. 6.6d. All shocks now appear sharper and the oscillations observed near the leading edge of the cowl has disappeared. Figs. 6.6f & 6.6h also show that discontinuity, as per the sensor, occurs in a narrower region.

A second round of r-adaptation is then applied in the exact similar fashion: shock areas are detected and used as input for the optimiser. The worst scaled Jacobian for this mesh is found to be $J_{s,\min} = 0.982$ after optimisation. Fig. 6.7b shows the final rr-adapted mesh which demonstrates further refinement of the shock regions. It is also noticed that the oblique shocks inside the intake past the throat have moved upstream due to the refinement of the oblique shocks located upstream of the throat. While the r-adapted mesh could not capture these downstream shocks, the rr-adapted mesh can. By using a two-step approach, it is possible to pull more mesh nodes together than when using a one-step approach.

This becomes even more obvious when looking at the plot of the Mach number in Fig. 6.8. The initial mesh largely overestimates the values of the Mach number past the throat ($x/L \approx 0.57$). This is mostly solved by r-adaptation although further



Initial

r-adapted

Mesh.

(a)

(b)

Mach number.

(c)

(d)

Sensor.

(e)

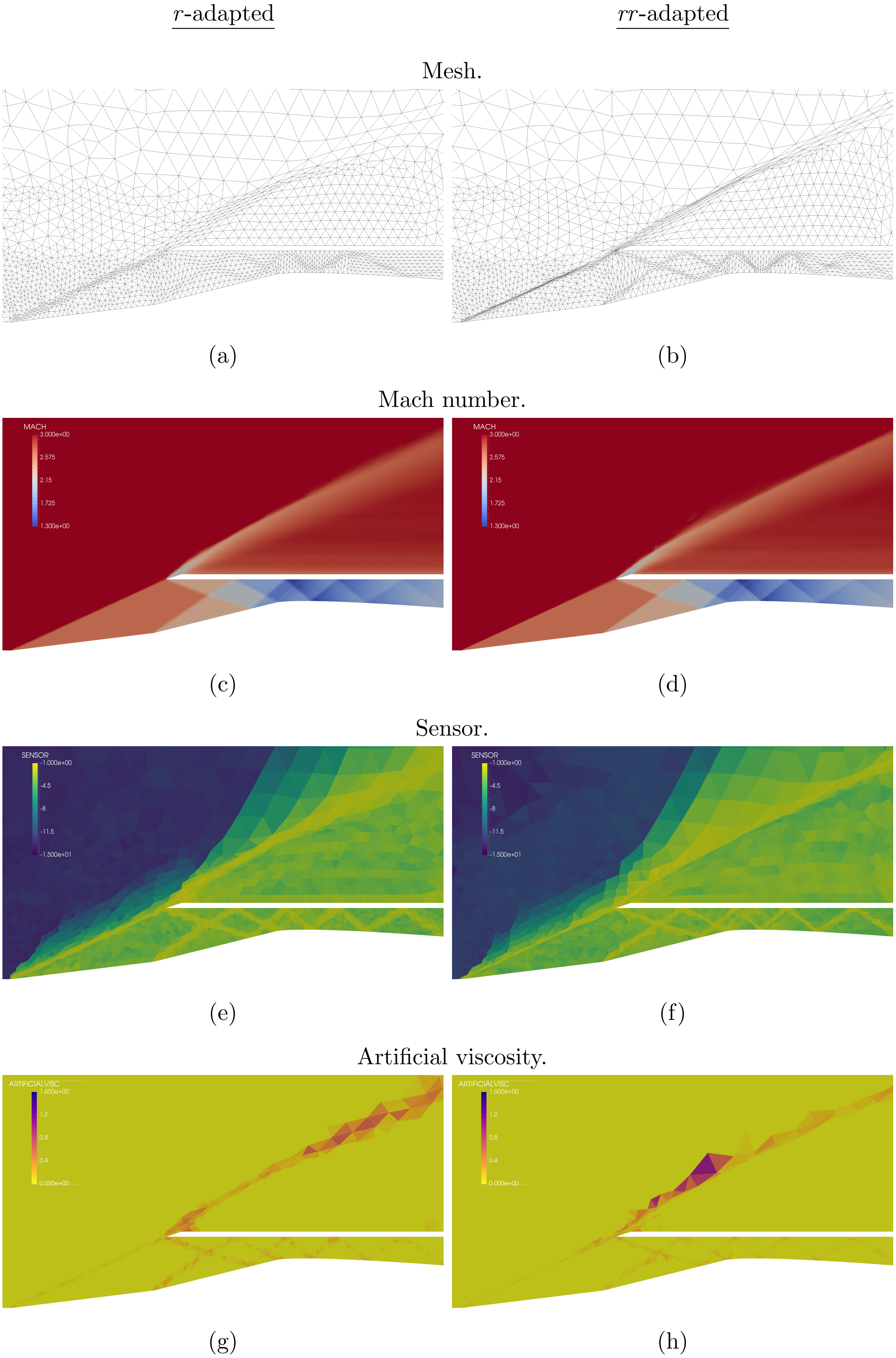
(f)

Artificial viscosity.

(g)

(h)

Figure 6.6: Mesh and fields for the intake before and after the first round of r-adaptation.



r-adapted

rr-adapted

Mesh.

(a)

(b)

Mach number.

(c)

(d)

Sensor.

(e)

(f)

Artificial viscosity.

(g)

(h)

Figure 6.7: Mesh and fields for the intake before and after the second round of r-adaptation.

improvement is obtained through rr-adaptation. This is due to the good resolution of upstream shocks through the clustering of DOF.

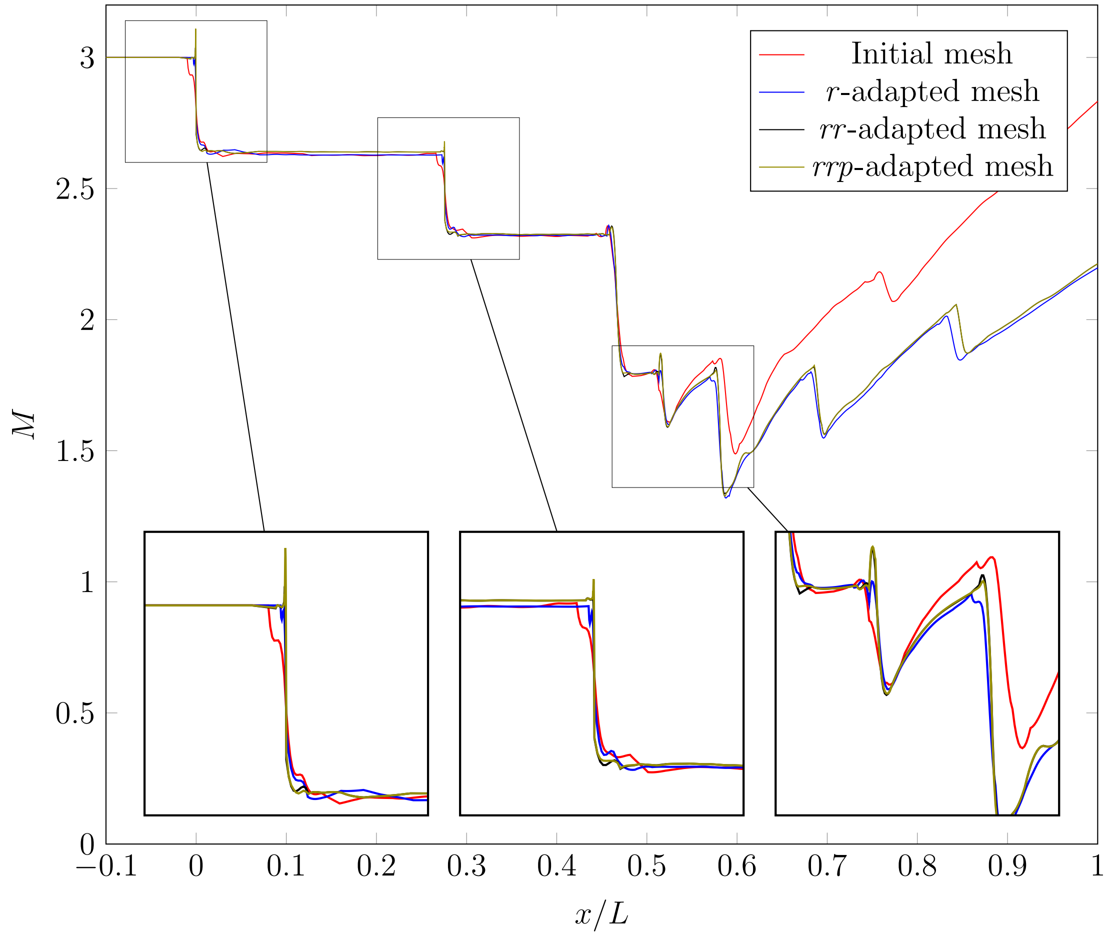


Figure 6.8: Plot of the Mach number on the lower surface.

p-adaptation

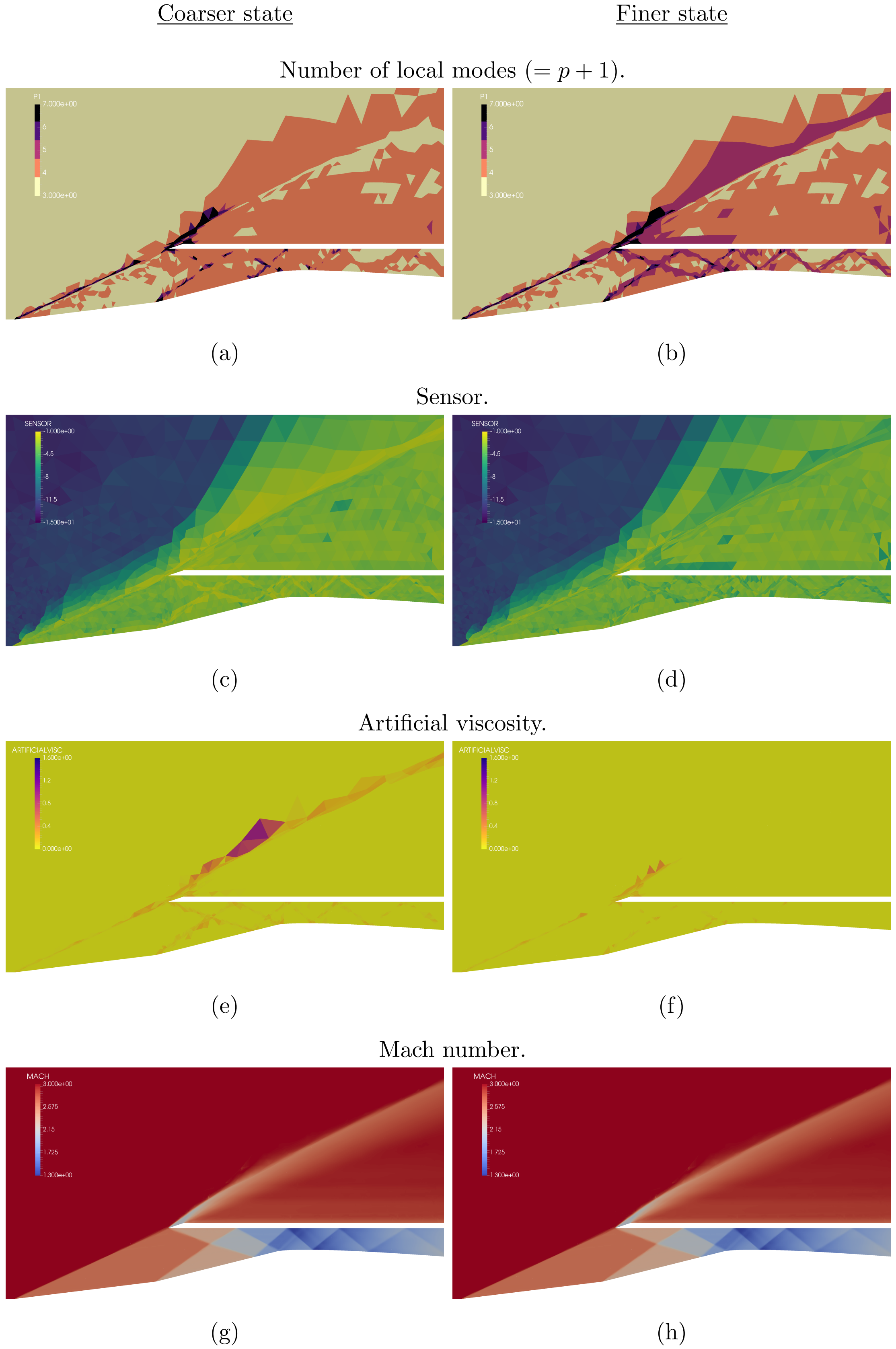
p-adaptation is now applied to the rr-adapted mesh. For this test case, only unrestricted p-adaptation is used where the local polynomial order inside elements is left free to change, even in shock areas. The field obtained at $p = 4$ in Sec. 6.4.2 is used as initial conditions and values of $p_{\text{min}} = 2$ and $p_{\text{max}} = 6$ are used. A sensor based on the density ρ and solver default values for the thresholds are again used.

First, it is observed that no steady state is achieved. Upon inspection, it is noticed that the system jumps back and forth between two states at each p-adaptation cycle. The two states correspond roughly to coarser and finer resolved fields. In the coarser resolved state, sensor values in shock areas are high. At the end of the p-adaptation cycle, these large sensor values trigger an increase in local polynomial order of a number of elements. Simulation goes on and the finer resolved state is obtained where sensor values are low. This in turn triggers a decrease in local polynomial order of the same elements, returning the system to the former coarser resolved field. This is shown in Fig. 6.9 with the coarser resolved state in Fig. 6.9g and the finer one in Fig. 6.9h. This behaviour is explained by a naive p-adaptation approach using simple sensor thresholds. The problem is highly non-linear and non-local and error from refining/coarsening regions propagates along characteristics. The non-adjoint nature of the refinement strategy is bound to produce this sort of behaviour.

Nevertheless it is observed that additional resolution in the form of higher local polynomials is found in sensible areas: in the shocks, inside the intake (especially in the throat) and right above the coil. The only very high polynomial orders are obtained in the shocks whereas smooth regions reach order $p = 3$ at most. The number of DOF for each simulation and state is shown in Table 6.2. Note that CPU times per time step are reported as run on a 16-core machine, once convergence is reached. Referring back to Fig. 6.8, that little difference in the solution can be seen from rr-adaptation to rrp-adaptation despite the decrease in number of DOF.

Table 6.2: Number of DOF and CPU time per time step at convergence for the intake.

Simulation	State	Number of DOF	CPU time (ms)
Uniform p		40 210	27
Unrestricted p-adaptation	Finer	39 527	74
Unrestricted p-adaptation	Coarser	36 696	69



Coarser state

Finer state

Number of local modes $(= p + 1)$.

(a)

(b)

Sensor.

(c)

(d)

Artificial viscosity.

(e)

(f)

Mach number.

(g)

(h)

Figure 6.9: Fields for the intake in its coarser and finer resolved states during unrestricted *p*-adaptation.

Chapter 7

Conclusions

The proposed techniques for mesh generation and adaptation are recalled and then discussed in this chapter. Limitations are also noted where appropriate, leading to recommendations for future work, as evaluated from the author's experience developing and working with these techniques.

7.1 Mesh generation

This thesis first covered the generation of quality initial meshes. In two dimensions, the proposed method aimed at creating valid quadrilateral block decompositions that can be refined into naturally curved, high-order meshes. This approach, however, does not extend well to three dimensions, in which case a semi-structured approach is used that partitions the domain into near-field and far-field regions that can be discretised separately in an easier way. The near-field partition, in particular, yields a structured boundary layer mesh with high aspect ratio, flow-aligned elements that are curved and valid. These two strategies, two- and three-dimensional, are discussed in what follows.

7.1.1 Two dimensions

The method presented in Ch. 3 is inspired by cross field methods. Among the possibilities for computing the guiding field, a Laplace problem is solved to smoothly propagate the boundary constraints to the interior. This approach differs from previous work in two ways: first high-resolution spectral/*hp* element methods are used to solve for the guiding field; second the location of the critical points and separatrices are not computed from crosses, but instead from the continuous guiding field. As a result, the method has advantages over the traditional techniques where numerical accuracy is the issue, but retains any disadvantages stemming from the original formulation of the problem. This means that there are both limitations and areas for further improvement.

Limitations

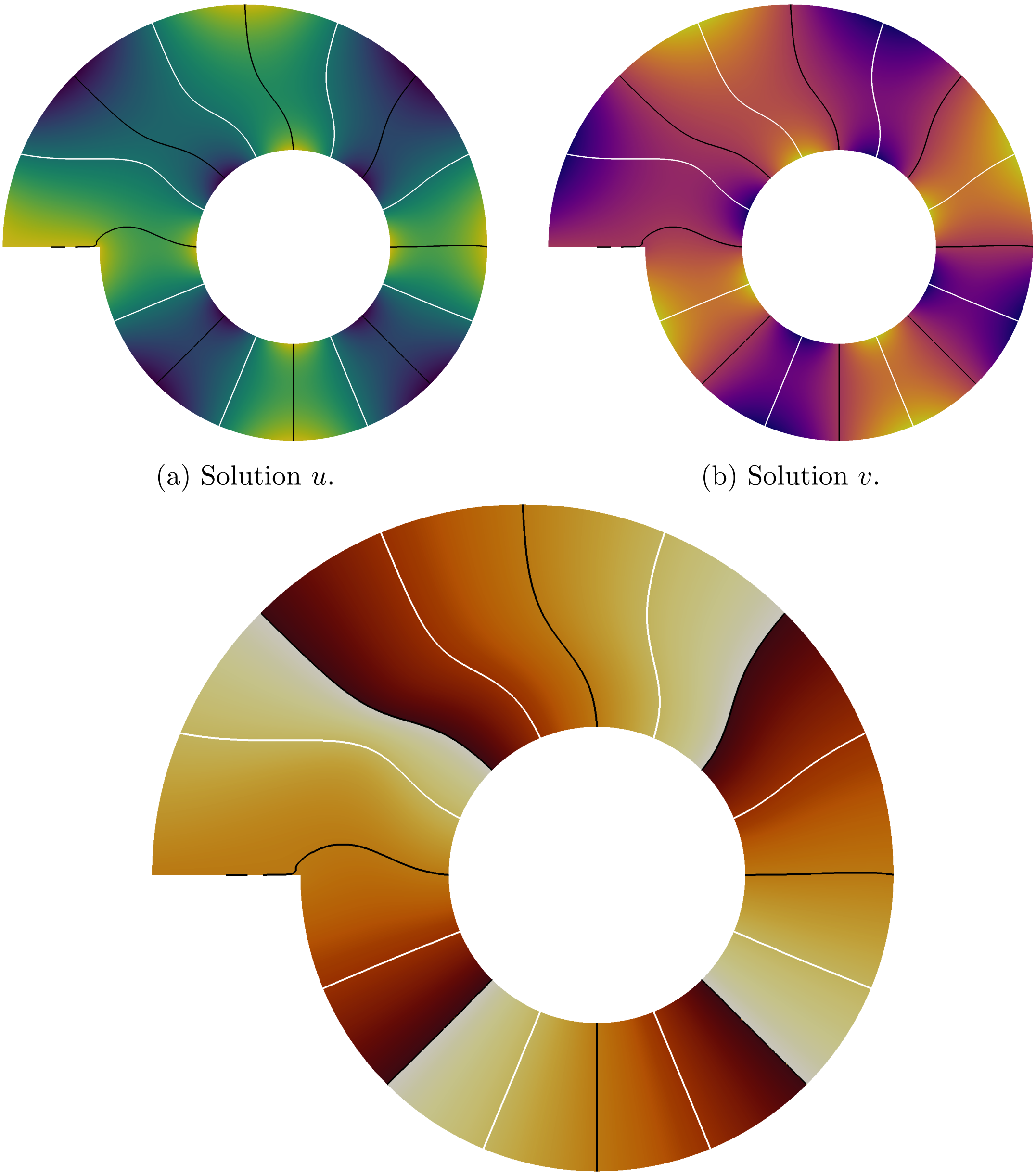
The method has the same limitations due to the formulation as cross field methods. One such limitation is the inability of cross fields to generate singular/critical points (and therefore irregular nodes) in domains where crosses are easily aligned with all nearby boundaries. Reference [25] documents a modified geometry of the nautilus (of Figs. 3.9, 3.15, 3.16) with a hole, for which a cross field is unable to generate a valid decomposition. Fig. 7.1 indeed shows that the field is aligned everywhere and that no critical point appears in the domain. Without critical points, however, a limit cycle is unavoidable and no valid quadrilateral decomposition can be obtained.

Also, the use of the high-order approximations mitigates, but does not eliminate numerical errors. Therefore, although the nautilus geometry is easily decomposed with the high-resolution approximation, it is still possible that spiral streamlines are generated if the accuracy is not high enough.

Finally the extension of this approach to three dimensional domains remains difficult. In three dimensions, crosses become *frames* but the corresponding arithmetic is currently lacking. Because this method is based on cross/frame fields, it will likewise inherit a lack of robustness in three dimensions.

Summary

A field guided method was developed to generate quadrilateral meshes for general two dimensional domains. Inspired by cross field mesh generation methods, the procedure consists of four steps: The first is to compute the guiding field on an existing triangular mesh. For that, a spectral/*hp* element method is used to compute a high-resolution approximation to the two Laplace problems for the guiding field. Either a CG or DG method is used depending on the angles formed along the physical boundaries. If the angles are not multiples of $\pi/2$, the DG formulation can handle the discontinuous BC in a discretisation consistent manner, without the need for *ad hoc* smoothing required by the CG formulation. Critical points in the field and their valences are then found by exploiting the local high-order polynomial approximations within each element. Streamlines are computed in the field, again, using the high-order solutions rather than crosses defined only at element corners. Finally the domain can be cut into elements of the desired size, or used as is.



(a) Solution u.

(b) Solution v.

(c) Computed ψ.

Figure 7.1: Guiding field on the holed nautilus: isocontours of $u = 0$ and $v = 0$ shown in white and black respectively.

An advantage of the approach is that it can generate meshes with naturally curved quadrilateral elements that do not need to be curved *a posteriori* to eliminate invalid elements. It avoids low-order errors that make it difficult to locate irregular nodes and accurately trace separatrices.

7.1.2 Three dimensions

Due to the lack of robust volume block decomposition methods based on frame fields, the approach of Ch. 3 is not explored in three dimensions. Instead, a semi-structured approach is proposed in Ch. 4 for generating high-order meshes with high aspect ratio elements to efficiently simulate boundary layer flows. It combines a linear mesh generator for hybrid linear meshes of hexahedral, prismatic and tetrahedral elements, based on a medial object approach for domain decomposition, and an *a posteriori* high-order mesh generator.

The domain is decomposed into near-field and far-field regions using a medial-object approach currently under development within *CADfix*. The medial object permits to design near-field regions that allow a boundary layer mesh which is "thicker" that those achievable by most commercially available mesh generators. The far-field region is discretised into tetrahedra. These topologies are simpler to obtain than a general multi-block decomposition, and yet they are sufficiently flexible to deal with reasonably complex geometries.

There are two main contributions of this chapter. The first one is the design of a concave and a concave/convex multi-normal topology for the near-field region in combination with a modified isoparametric approach that produces hybrid meshes with hexahedral elements at junctions. These meshes have a higher quality and fewer elements than those corresponding to direct topologies. The second contribution is the incorporation of periodic surfaces both in medial-object decomposition and mesh generation. This permits the numerical treatment of flow simulations incorporating periodic BC.

The proposed method is robust for both linear and, to some extent, high-order meshing, but its ability to produce high-order meshes of very good quality depends strongly on the quality of the linear mesh and of the distortions induced by the CAD surface mappings. The major contributing factors to this problem are the coarseness of the linear mesh required and the higher sensitivity of high-order algorithms to distortions in the mappings defining the CAD curves and surfaces. However, through the use of the semi-structured approach proposed here, the rate of success of producing high-order meshes with complex geometries has been significantly improved.

7.2 Mesh adaptation

A novel strategy has been presented for adaptive simulations, based on a combination of both r- and p-adaptation. The proof-of-concept work applied here takes advantage of both strategies in different manners, as appropriate for the simulation of compressible flows containing shocks. Mesh movement required for r-adaptation is achieved through the use of the variational optimisation strategy, using the combination of a local discontinuity sensor and a target element size in order to effectively cluster DOF in the presence of shocks and more sharply simulate their features. At the same time, a p-adaptation technique is applied in the rest of the domain in order to benefit from the spectral rate of convergence of high-order discretisations for smooth solutions. The simulation is effectively stabilised through the use of an artificial diffusion term, again using the local discontinuity sensor.

7.2.1 *r*-adaptation

A variational framework for moving meshes, referred to here as r-adaptation, was presented in Ch. 5. It is derived from a variational formulation originally designed for high-order mesh optimisation. The technique has been modified to achieve adaptation by manipulating a target element. The optimiser, by minimisation of the energy functional, deforms the mesh to best fit the target element shape and size defined by the user.

The framework is first demonstrated on toy analytical problems before being driven by an error estimator. These examples show promising results with regards to the feasibility of this approach. Most importantly, the framework retains the benefits of the original work, in particular its scalability. Secondly, the ability to conform to complex CAD surfaces and curves whilst permitting nodes to slide across them is clearly important in the context of this work, where adaptation is required at or near solid surfaces. This functionality can be difficult to achieve in other mesh deformation techniques, particularly those that require the solution of a PDE system of an appropriate solid body model.

7.2.2 *rp*-adaptation

The rp-adaptation strategy proposed in Ch. 6 exhibits a number of benefits from a computational perspective, as seen in the presented results, where the canonical NACA 0012 test case and a more challenging supersonic intake have been examined. The main benefit of this dual-adaptive technique is the possibility to significantly reduce the number of DOF required to resolve a given simulation, when compared to a uniformly refined grid or using solely r-adaptation. Table 6.1 shows that, for the various p-adaptation strategies considered, the error when compared to a very fine solution remains roughly the same, whilst the simulation requires only 50% of the DOF of the initial simulation. This has important consequences from the perspective of computational efficiency, since a significant reduction in the number of DOF will lead to a reduction in runtime. Likewise, the cost of operations per DOF is reduced as the polynomial order decreases, which offers the opportunity to further reduce computational cost. The rp-adaptation technique therefore permits an effective balance to be achieved between the attained error and simulation expense.

From the context of more general conclusions of these results, it is demonstrated that care must be taken when selecting a p-adaptation strategy. In particular, the NACA 0012 simulations demonstrate that p-coarsening can have important negative effects on the solution for minimal computational gains. Additionally, the supersonic intake exhibits a complex shock pattern. Because of the complexity and strength of the reflecting shocks, it is shown that multiple r-adaptation steps are not only possible but desirable. Despite the lack of nodes to redistribute inside the intake, sufficient mesh deformation is achieved to better capture the different shocks.

Although the overall strategy has been shown to be effective, it is important to emphasise that some of the benefits highlighted in this work can be attributed to this particular implementation of the r-adaptation technique. Importantly, the use of the variational framework yields several advantages. In particular, the use of a target element size allows the grid to deform in an anisotropic manner within restricted regions of the domain. Even when the deformation is substantial, this still permits a valid grid to be obtained, as shown in Fig. 6.1.

7.3 Recommendations for future work

This final section presents some recommendations towards future work related to the techniques presented in this thesis, in particular the field guided quadrilateral mesh generation and rp-adaptation. These two methods are proofs-of-concept requiring more improvements and analysis. The hybrid approach to three dimensional mesh generation, on the other hand, has been discussed above and is considered, by the author, to have have reached a certain level of maturity. Refinements are certainly possible but to a lesser extent.

7.3.1 Quadrilateral meshes

First the limitations of the guiding field approach to quadrilateral mesh generation point the way towards possible improvements to both the formulation and solution of the problem. Modifications to the guiding field formulation might be used to affect the number and location of critical points and even the existence of a valid quadrilateral decomposition. Modifications to the algorithm could be made to control such problems.

The guiding field could be adjusted by adding a forcing term to Eqs. (3.4), thus solving a Poisson rather than a Laplace problem, which would allow the guiding field to be modified in several ways. For example, critical points could be added [13, 14] so that a valid decomposition can be obtained for problems like the holed nautilus. The reader should note that the midpoint splitting the degenerate triangle in Fig. 3.11 is also an example of placing an extra irregular point in the decomposition. For the nautilus with hole, a hand sketch shows that adding one 3- and one 5-valent irregular nodes is sufficient. Existing critical points might also be moved away from boundaries to avoid thin blocks in the decomposition using the idea of control functions used in elliptic grid generation [74]. Finally a forcing term would allow one to generate a guiding field that partially aligns with a metric field (incl. from a three dimensional surface), such as was demonstrated in [25].

Performance improvements can also be made. Streamline integration can be performed in reference space [21], akin to the location and analysis of the critical points. This would not only be computationally cheaper, it would also allow one to more easily handle geometries with large differences in scales. Where small scales are present, small elements are typically automatically generated for the background mesh to represent the geometry boundaries accurately. By performing all of the analysis of the guiding field in reference space, different scales would be automatically handled and better performance might be achieved. Streamline integration with a symplectic integrator might also be implemented to further mitigate the appearance of numerically created spirals. Finally adaptive ODE time integration methods could further enhance the accuracy of the streamline integration.

An extension of this methodology to volume block decompositions would represent a breakthrough in cross field based techniques. It would, however, require a reformulation of the problem based on frame arithmetic, which is currently lacking. Promising work in reference [63] points to a nine variable problem to solve.

7.3.2 *rp*-adaptation

A number of directions for potential future work are also clear in the area of *rp*-adaptation. An extension of this method to transient flows, especially with moving shocks, would constitute an interesting application of this *rp*-adaptation strategy. The variational moving mesh framework would be able to track shocks throughout the simulation without the need to generate a new mesh. With preserved mesh connectivities, the system of equations would not need to be re-built at each adaptation step. This is especially desirable on large meshes and large HPC[1]-based simulations where I/O and inter-node communication can incur significant expense.

Several developments would, however, be needed for the approach to work in transient simulations. Most importantly, the compressible flow solver would need an arbitrary Lagrangian-Eulerian formulation to handle mesh movements and avoid any kind of interpolation that would reduce the accuracy of the simulation. A more thorough analysis of the current shock sensor and its link to shock detection will be needed to ensure robustness in both the addition of artificial diffusion for shock capturing and the *r*-adaptation procedure. More generally, the workflow will have to be redesigned to account for the unsteadiness of the simulation.

[1] HPC: high-performance computing.

Bibliography

[1] F. Alauzet and A. Loseille. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Computer-Aided Design*, 72:13–39, 2016.

[2] W.E. Anderson, N.D. Wong, and M. Field. Experimental Investigation of a Large-Scale, Two-Dimensional, Mixed-Compression Inlet System - Performance at Design Conditions, M = 3.0. Technical report, NASA Ames Research Center, Mountain View, CA, USA, 1970.

[3] C.G. Armstrong, H.J. Fogg, C.M. Tierney, and T.T. Robinson. Common Themes in Multi-block Structured Quad/Hex Mesh Generation. *Procedia Engineering*, 124:70–82, 2015.

[4] G.E. Barter and D.L. Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *Journal of Computational Physics*, 229(5):1810–1827, 2010.

[5] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1996.

[6] T.D. Blacker, M.B. Stephenson, and S. Canann. Analysis automation with paving: A new quadrilateral meshing technique. *Advances in Engineering Software and Workstations*, 13(5-6):332–337, 1991.

[7] H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, 5:362–380, 1967.

[8] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin. Quad-Mesh Generation and Processing: A Survey. *Computer Graphics Forum*, 32(6):51–76, 2013.

[9] R. Bouffanais. Advances and challenges of applied large-eddy simulation. *Computers & Fluids*, 39(5):735–738, 2010.

[10] J.H. Bucklow. 3D medial object computation using a domain Delaunay triangulation. In *Medial Object Technology Workshop*, 2014.

[11] J.H. Bucklow, R. Fairey, and M.R. Gammon. An automated workflow for high quality CFD meshing using the 3D medial object. In *23rd AIAA Computational Fluid Dynamics Conference*, Denver, CO, USA, 2017. American Institute of Aeronautics and Astronautics. AIAA 2017-3454.

[12] C.J. Budd, W. Huang, and R.D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:1–131, 2009.

[13] G. Bunin. A Continuum Theory for Unstructured Mesh Generation in Two Dimensions, 2006, arXiv:cs/0609078.

[14] G. Bunin. Towards Unstructured Mesh Generation Using the Inverse Poisson Problem, 2008, arXiv:physics.comp-ph/0802.2399.

[15] A. Burbeau and P. Sagaut. A dynamic p-adaptive Discontinuous Galerkin method for viscous flow with shocks. *Computers & Fluids*, 34(4-5):401–417, 2005.

[16] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimisation. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[17] C.D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R.M. Kirby, and S.J. Sherwin. Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications*, 192:205–219, 2015.

[18] S. Chow, G. Zilliac, and P. Bradshaw. Mean and turbulence measurements in the near field of a wingtip vortex. *AIAA Journal*, 35(10):1561–1567, 1997.

[19] B. Cockburn and C.W. Shu. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.

[20] J. Cohen, J. Marcon, M. Turner, C.D. Cantwell, S.J. Sherwin, J. Peiró, and D. Moxey. Simplifying high-order mesh generation for computational scientists. In M. Atkinson and S. Gesing, editors, *Proceedings of the 10th International Workshop on Science Gateways*, Edinburgh, Scotland, 2019. CEUR-WS.org.

[21] G. Coppola, S.J. Sherwin, and J. Peiró. Nonlinear Particle Tracking for High-Order Elements. *Journal of Computational Physics*, 172(1):356–386, 2001.

[22] D. Ekelschot, M. Ceze, A. Garai, and S.M. Murman. Robust metric aligned quad-dominant meshing using Lp centroidal Voronoi tessellation. In *2018 AIAA Aerospace Sciences Meeting*, Reston, Virginia, 2018. American Institute of Aeronautics and Astronautics.

[23] D. Ekelschot, M. Ceze, S.M. Murman, and A. Garai. Parallel high-order anisotropic meshing using discrete metric tensors. In *AIAA Scitech 2019 Forum*, Reston, Virginia, 2019. American Institute of Aeronautics and Astronautics.

[24] D. Ekelschot, D. Moxey, S.J. Sherwin, and J. Peiró. A p-adaptation method for compressible flow problems using a goal-based error indicator. *Computers & Structures*, 181:55–69, 2017.

[25] H.J. Fogg, C.G. Armstrong, and T.T. Robinson. Automatic generation of multiblock decompositions of surfaces. *International Journal for Numerical Methods in Engineering*, 101(13):965–991, 2015.

[26] Free Software Foundation, Inc. lpsolve 5.5. `lpsolve.sourceforge.net`, 2019. software.

[27] P.J. Frey and F. Alauzet. Anisotropic mesh adaptation for CFD computations. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):5068–5082, 2005.

[28] A. Gargallo-Peiró, X. Roca, J. Peraire, and J. Sarrate. Distortion and quality measures for validating and generating high-order tetrahedral meshes. *Engineering with Computers*, 31(3):423–437, 2015.

[29] N.J. Georgiadis, D.P. Rizzetta, and C. Fureby. Large-Eddy Simulation: Current Capabilities, Recommended Practices, and Future Research. *AIAA Journal*, 48(8):1772–1784, 2010.

[30] F. Hindenlang. *Mesh Curving Techniques for High Order Parallel Simulations on Unstructured Meshes*. PhD thesis, University of Stuttgart, 2014.

[31] W. Huang and R.D. Russell. *Adaptive moving mesh methods*. Springer, 2011.

[32] International TechneGroup, Ltd. Private Communication: Experimental implementation of a cross field based quad subdivision scheme in CADfix, 2019.

[33] International TechneGroup, Ltd. CADfix: CAD translation, healing, repair, and transformation. `iti-global.com/cadfix`, 2019. software.

[34] M.K. Jain and S. Mittal. Euler flow in a supersonic mixed-compression inlet. *International Journal for Numerical Methods in Fluids*, 50(12):1405–1423, 2006.

[35] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2nd edition, 2005.

[36] D.A. Kopriva. *Implementing Spectral Methods for Partial Differential Equations*. Scientific Computation. Springer, 2009.

[37] N. Kowalski, F. Ledoux, and P. Frey. A PDE based approach to multi-domain partitioning and quadrilateral meshing. In *Proceedings of the 21th International Meshing Roundtable*, pages 137–154. Springer Berlin Heidelberg, 2013.

[38] J.E. Kozdon, E.M. Dunham, and J. Nordström. Simulation of Dynamic Earthquake Ruptures in Complex Geometries Using High-Order Finite Difference Methods. *Journal of Scientific Computing*, 55(1):92–124, 2013.

[39] T.S. Li, R.M. McKeag, and C.G. Armstrong. Hexahedral meshing using midpoint subdivision and integer programming. *Computer Methods in Applied Mechanics and Engineering*, 124(1-2):171–193, 1995.

[40] Y. Li, S. Premasuthan, and A. Jameson. Comparison of Adaptive h and p Refinements for Spectral Difference Methods. In *40th Fluid Dynamics Conference and Exhibit*, Reston, VA, USA, 2010. American Institute of Aeronautics and Astronautics.

[41] X. Liang, M. S. Ebeida, and Y. Zhang. Guaranteed-quality all-quadrilateral mesh generation with feature preservation. *Computational Methods in Applied Mechanics and Engineering*, 199(29-32):2072–2083, 2010.

[42] J.-E.W. Lombard, D. Moxey, S.J. Sherwin, J.F.A. Hoessler, S. Dhandapani, and M.J. Taylor. Implicit large-eddy simulation of a wingtip vortex. *AIAA Journal*, 54(2):506–518, 2016.

[43] X.-J. Luo, M.S. Shephard, R.M. O'Bara, R. Nastasia, and M.W. Beall. Automatic p-version mesh generation for curved domains. *Engineering with Computers*, 20(3):273–285, 2004.

[44] J. Marcon, G. Castiglioni, D. Moxey, S.J. Sherwin, and J. Peiró. *rp*-adaptation for compressible flows, 2019, arXiv:physics.comp-ph/1909.10973. under review.

[45] J. Marcon, D.A. Kopriva, S.J. Sherwin, and J. Peiró. A high resolution PDE approach to quadrilateral mesh generation. *Journal of Computational Physics*, 399C:108918, 2019.

[46] J. Marcon, D.A. Kopriva, S.J. Sherwin, and J. Peiró. Naturally curved quadrilateral mesh generation using an adaptive spectral element solver. In *Proceedings of the 28th International Meshing Roundtable*, Buffalo, NY, USA, 2019. in press.

[47] J. Marcon, J. Peiró, D. Moxey, N. Bergemann, H. Bucklow, and M.R. Gammon. A semi-structured approach to curvilinear mesh generation around streamlined bodies. In *AIAA Scitech 2019 Forum*, Reston, Virginia, 2019. American Institute of Aeronautics and Astronautics.

[48] J. Marcon, M. Turner, D. Moxey, S.J. Sherwin, and J. Peiró. A variational approach to high-order r-adaptation. In *26th International Meshing Roundtable*, 2017.

[49] J. Marcon, M. Turner, J. Peiró, D. Moxey, C. Pollard, H. Bucklow, and M.R. Gammon. High-order curvilinear hybrid mesh generation for CFD simulations. In *2018 AIAA Aerospace Sciences Meeting*, Reston, Virginia, 2018. American Institute of Aeronautics and Astronautics.

[50] D. Moxey, C.D. Cantwell, Y. Bao, A. Cassinelli, G. Castiglioni, S. Chun, E. Juda, E. Kazemi, K. Lackhove, J. Marcon, G. Mengaldo, D. Serson, M. Turner, H. Xu, J. Peiró, R.M. Kirby, and S.J. Sherwin. Nektar++: Enhancing the capability and application of high-fidelity spectral/*hp* element methods. *Computer Physics Communications*, 107110, 2019. in press.

[51] D. Moxey, C.D. Cantwell, G. Mengaldo, D. Serson, D. Ekelschot, J. Peiró, S.J. Sherwin, and R.M. Kirby. Towards p-Adaptive Spectral/hp Element Methods for Modelling Industrial Flows. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pages 63–79. Springer, Cham, 2017.

[52] D. Moxey, D. Ekelschot, Ü. Keskin, S.J. Sherwin, and J. Peiró. A Thermoelastic Analogy for High-order Curvilinear Meshing with Control of Mesh Validity and Quality. *Procedia Engineering*, 82:127–135, 2014.

[53] D. Moxey, M.D. Green, S. J. Sherwin, and J. Peiró. *On the generation of curvilinear meshes through subdivision of isoparametric elements*, pages 203–215. Springer, 2015.

[54] D. Moxey, M.D. Green, S.J. Sherwin, and J. Peiró. An isoparametric approach to high-order curvilinear boundary-layer meshing. *Computer Methods in Applied Mechanics and Engineering*, 283:636–650, 2015.

[55] F. Naddei, M. de la Llave Plata, and V. Couaillier. A comparison of refinement indicators for p-adaptive discontinuous Galerkin methods for the Euler and Navier-Stokes equations. In *2018 AIAA Aerospace Sciences Meeting*, page 0368, Reston, Virginia, 2018. American Institute of Aeronautics and Astronautics.

[56] F. Naddei, M. de la Llave Plata, V. Couaillier, and F. Coquel. A comparison of refinement indicators for p-adaptive simulations of steady and unsteady flows using discontinuous Galerkin methods. *Journal of Computational Physics*, 376:508–533, 2019.

[57] NASA. Common Research Model. `commonresearchmodel.larc.nasa.gov`, 2017. file.

[58] Open Cascade. Open Cascade Technology. `opencascade.com/content/open-source-core-technology`, 2019. software.

[59] M.A. Park, A. Loseille, J. Krakos, T.R. Michal, and J.J. Alonso. Unstructured Grid Adaptation: Status, Potential Impacts, and Recommended Investments Towards CFD 2030. In *46th AIAA Fluid Dynamics Conference*, Reston, Virginia, 2016. American Institute of Aeronautics and Astronautics.

[60] P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin method. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, US, 2006. American Institute of Aeronautics and Astronautics. AIAA 2006-112.

[61] M.J. Pierzga and J.R. Wood. Investigation of the Three-Dimensional Flow Field Within a Transonic Fan Rotor: Experiment and Analysis. *ASME Journal of Engineering for Gas Turbines and Power*, 107(2):436–449, 1985.

[62] R. Poya, R. Sevilla, and A.J. Gil. A unified approach for a posteriori high-order curved mesh generation using solid mechanics. *Computational Mechanics*, 58(3):457–490, 2016.

[63] N. Ray, D. Sokolov, and B. Lévy. Practical 3D frame field generation. *ACM Transactions on Graphics*, 35(6):1–9, 2016.

[64] J.-F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Béchet, E. Marchandise, C. Geuzaine, and T. Mouton. A frontal Delaunay quad mesh generator using

the l^∞ norm. *International Journal For Numerical Methods In Engineering*, 94(5):494–512, 2013.

[65] F. Renac, M. de la Llave Plata, E. Martin, J.-B. Chapelier, and V. Couaillier. Aghora: A High-Order DG Solver for Turbulent Flow Simulations. In *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, volume 128, pages 315–335. Springer Verlag, 2015.

[66] O. Sahni, X.J. Luo, K.E. Jansen, and M.S. Shephard. Curved boundary layer meshing for adaptive viscous flow simulations. *Finite Elements in Analysis and Design*, 46(1-2):132–139, 2010.

[67] R. Schneiders. Algorithms for Quadrilateral and Hexahedral Mesh Generation. Technical report, MAGMA Gießereitechnologie GmbH.

[68] R. Schneiders, R. Schindler, and F. Weiler. Octree-based Generation of Hexahedral Element Meshes. In *Proceedings of the 5th International Meshing Roundtable*, pages 205—-215, 1996.

[69] D. Sheehy. *Medial surface computation using a domain Delaunay triangulation.* PhD thesis, Queen's University of Belfast, 1994.

[70] S.J. Sherwin and J. Peiró. Mesh generation in curvilinear domains using high-order elements. *International Journal for Numerical Methods in Engineering*, 53(1):207–223, 2002.

[71] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D.J. Mavriplis. CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences. Technical report, 2014.

[72] M.L. Staten, R.A. Kerr, S.J. Owen, T.D. Blacker, M. Stupazzini, and K. Shimada. Unconstrained plastering-Hexahedral mesh generation via advancing-front geometry decomposition. *International Journal for Numerical Methods in Engineering*, 81:135–171, 2009.

[73] T. Tam and C.G. Armstrong. Finite element mesh control by integer programming. *International Journal for Numerical Methods in Engineering*, 36(15):2581–2605, 1993.

[74] J.F. Thompson, B.K. Soni, and N.P. Weatherill. *Handbook of Grid Generation.* CRC Press, 1999.

[75] B.H.V. Topping, J. Muylle, P. Ivanyi, R. Putanowicz, and B. Cheng. *Finite Element Mesh Generation.* Saxe-Coburg Publications, 2002.

[76] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, Berlin, Heidelberg, 2009.

[77] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous Topology Simplification of Planar Vector Fields. In *Proceedings of the Conference on Visualization '01*, pages 159–166, Washington, DC, USA, 2001. IEEE Computer Society.

[78] M. Turner. *High-order mesh generation for CFD solvers.* Phd thesis, Imperial College London, 2018.

[79] M. Turner, D. Moxey, S.J. Sherwin, and J. Peiró. Automatic generation of 3D unstructured high-order curvilinear meshes. In *VII European Congress on Computational Methods in Applied Sciences and Engineering*, Crete Island, Greece, 2016. ECCOMAS.

[80] M. Turner, J. Peiró, and D. Moxey. Curvilinear mesh generation using a variational framework. *Computer-Aided Design*, 103:73–91, 2018.

[81] J.C. Vassberg, M.A. DeHaan, S.M. Rivers, and R.A. Wahls. Development of a Common Research Model for Applied CFD Validation Studies. In *26th AIAA Applied Aerodynamics Conference*, Honolulu, HI, USA, 2008. American Institute of Aeronautics and Astronautics. AIAA 2008-6919.

[82] J.C. Vassberg and A. Jameson. In Pursuit of Grid Convergence for Two-Dimensional Euler Solutions. *Journal of Aircraft*, 47(4):1152–1166, 2010.

[83] R. Viertel and B. Osting. An Approach to Quad Meshing Based on Harmonic Cross-Valued Maps and the Ginzburg–Landau Theory. *SIAM Journal on Scientific Computing*, 41(1):A452–A479, 2019.

[84] R. Viertel, M. Staten, and B. Osting. Toward a paver replacement. In *26th International Meshing Roundtable*, 2017.

[85] P.E. Vincent and A. Jameson. Facilitating the Adoption of Unstructured High-Order Methods Amongst a Wider Community of Fluid Dynamicists. *Mathematical Modeling Natural of Phenomena*, 6(3):97–140, 2011.

[86] Z.J. Wang, K.J. 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. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[87] H. Xu, C.D. Cantwell, C. Monteserin, C. Eskilsson, A.P. Engsig-Karup, and S.J. Sherwin. Spectral/hp element methods: Recent developments, applications, and perspectives. *Journal of Hydrodynamics*, 30(1):1–22, 2018.

[88] M. Yano. *An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes*. PhD thesis, Massachusetts Institute of Technology, 2012.