

IMPERIAL

Imperial College London
Department of Aeronautics

High-Order Mesh Generation and Adaptation for Complex Industrial Geometries

Kaloyan S. Kirilov

1st Supervisor: Prof. Joaquim Peiró
Department of Aeronautics
Imperial College London

2nd Supervisor: Prof. David Moxey
Department of Engineering
King's College London

3st Supervisor: Prof. Spencer Sherwin
Department of Aeronautics
Imperial College London

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy at Department of Aeronautics
the Diploma of Imperial College, July 2025

Abstract

High-order methods have been a standard for large eddy (LES) and direct numerical simulations (DNS) in academia and are now emerging as prime candidates for high-fidelity industrial CFD. A key bottleneck for this industrialization are the high-order meshing tools that automatically and robustly generate coarse high-quality body-fitted curvilinear meshes with sufficient resolution for resolving the flow features. Therefore, this thesis addresses the challenge through two complementary approaches: the robust generation of coarse curved meshes and mesh modification techniques allowing for efficient Adaptive Mesh Refinement (AMR).

In the first part, we propose a method for high-order mesh generation from third-party meshes. At its core is an approach for reconstructing the link between the mesh and the CAD model, which allows to leverage the robustness and flexibility of third-party straight-sided mesh generators with in-house state-of-the-art high-order mesh curving and optimisation techniques. We further address regions initially unsuitable for mesh curving through high-order mesh refinement and reformulate the long standing isoparametric prism layer splitting approach to allow generic boundary layer meshes. With these, we demonstrate robust generation of coarse curvilinear meshes for complex geometries such as F1 front wings and a full aircraft in high-lift configuration.

In the second part, we present a novel implementation of high-order conformal h-adaptation with predefined in the reference space maps in 2D and 3D. Through canonical examples, we demonstrate that it maintains mesh curvature and validity and often improves mesh quality. By combining the h-adaptation with a feature-based error indicator, we show that a very coarse, underresolved mesh can be adapted and capture the correct transition mechanisms. Finally, we demonstrate the potential of a true spectral/hp method, resolving accurately two inviscid test cases with combinations of h, r and p AMR, with the final h-r-p adaptation requiring hundreds of times fewer DoF than uniform refinement.

Acknowledgements

In the past four years, I was blessed to have an army of people who cared about my academic journey, the success of this project and myself. And naming all of you would be close to impossible, but I will try.

First of all, I would like to thank Joaquim Peiro, who went above and beyond to help me understand all intricacies in the world of meshing, CFD and academia, who was happy to stay for hours brainstorming my next wild idea with with patience, care and wisdom. Of course, neither of these would have come to reality if it were not Spencer Sherwin and David Moxey, who would solve weeks of struggle on the code with a single sentence - "Look at this line" and 'voila'. I feel incredibly fortunate to have the opportunity to learn and be guided by you during this PhD.

To Julien, Adam, Thanos and everyone in the CFD team who gave me the chance to have a glimpse into the world of motorsport. Learning from your immense experience and knowledge was a rare privilege, but watching this endless energy and unique team spirit making even the hardest fun, was absolutely unique. To Ramon Codina and Xevi Roca for hosting me at CIMNE and BSC during my secondment.

To Alex, Parv, Henrik (the London Mafia), Jingtian, you were not just my colleagues in the group, but a great team always ready to help each other with their problems, with research, true friendship and much, much more. To my girlfriend, who has been a steady support through the long nights and difficult days thank you for your patience and the relentless nudge to "cover the basics". And of course, I cannot miss the Aero friends: Kazuki, master Guglielmo, Chi Hin and all of the 141, 141A gang - it was a fun journey, I wouldn't change it for anything. I have found the greatest people around.

And last, while this marks the end of my PhD, I would like to thank my parents, who were supporting me not just in the past four years, but gave everything for me through the journey from Varna, Holland, UK and beyond for decades. I cannot believe how lucky I am with you.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 955923. Computational resources were supported in part through the Imperial College Research Computing Service (DOI: 10.14469/hpc/2232).

Declaration of Originality

The work presented in this thesis was conducted by the author as part of the requirements for the degree of Doctor of Philosophy in the Department of Aeronautics at Imperial College London. The research and results included in this thesis are solely the authors' unless it is specifically acknowledged. This work was part of the author's PhD and has not been previously submitted for other qualifications.

Kaloyan S. Kirilov

31 July 2025

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Contents

Abstract	1
Acknowledgements	3
1 Introduction	21
I High-Order Mesh Generation	24
2 Literature Study & Research Problem	25
2.1 The High-Order Element	25
2.2 High-Order Mesh Generation and its Challenges	27
2.2.1 Mesh Quality for Coarse Highly Stretched Viscous Layer Meshes	27
2.2.2 Geometrical Accuracy & Sources of Geometrical Perturbations	29
2.2.3 CAD Quality	32
2.2.4 Geometrical Complexity and Element Flexibility	34
2.2.5 Automation and Integration in Existing CFD Processes	34
2.3 Current State of the Art	35
2.4 Research Objectives	36
3 Proposed Methodology	37
3.1 High-Order Mesh Generation with NekMesh	37
3.2 Proposed Methodology	39
3.3 Workflow Implementation	41
3.3.1 Linear Mesh Generation	41

3.3.2	CAD-reconstruction	42
3.3.3	Mesh Curving and Surface Quality	47
3.4	Surface Mesh Optimizations	48
3.4.1	Assessing Geometrical Accuracy through CAD-Reconstruction	49
3.4.2	Variational Optimizer (Volume Optimization)	51
3.5	An Improved Isoparametric Boundary-layer Splitting	52
4	Applications to Industrial Geometries	53
4.1	Consistency - A Semi-spherical Geometry Test Case	53
4.2	Imperial Front Wing	56
4.2.1	Some Limitations And Important Notes of the methodology	64
5	Advanced Meshing Concepts	68
5.1	Closing the loop of Automation	68
5.2	Generic Viscous Layers	73
6	Conclusions and Future Work	78
6.1	Conclusions	78
6.2	Current & Future Work	81
II	Adaptive Mesh Refinement	84
7	Literature Review & Research Questions	85
7.1	Motivation	85
7.2	Mesh Adaptation Overview	87
7.3	CFD Solver	88
7.3.1	CG Discretization - Incompressible Navier-Stokes	88
7.3.2	DG Discretization - Compressible Navier-Stokes	89
7.4	Error Indicators and Mesh Metrics	90
7.4.1	Feature-based Error Indicators	90
7.4.2	Truncation Error estimators	90

7.4.3	Goal Oriented Error Estimation	91
7.4.4	Metrics and Adaptation Drivers	91
7.4.5	Novel Intelligent Error Indicators and Adaptation Drivers	93
7.5	Adaptation Mechanics and Mesh Modifications	93
7.6	Interpolation and Solution Transfer	98
7.7	The need for hp-adaptation	99
7.7.1	P-adaptation for smooth flow regions	99
7.7.2	h and r-adaptation for discontinuous flow features	100
7.7.3	State-of-the-art	100
7.8	Research Objectives	101
8	High-Order Curvilinear h-adaptation	102
8.1	High-Order Isoparametric Splitting	103
8.1.1	h-adaptation driver	104
8.1.2	2D Isoparametric Splitting	104
8.1.3	3D Isoparametric Splitting	105
8.2	Quality Improving Operators	107
8.2.1	High-Order Swapping	108
8.2.2	Mesh Smoothing	109
8.3	2D - Verification & Validation	110
8.3.1	Uniform Adaptation, Geometrical Accuracy, Mesh Quality and Performance	110
8.3.2	Conformal h-adaptation	112
8.4	3D - Verification & Validation	113
8.4.1	Uniform Adaptation, Geometrical Accuracy, Mesh Quality and Performance	113
8.4.2	Conformal h-adaptation	115
9	Towards AMR for Highly Separated Incompressible Flows	117
9.1	The choice of Feature-Based Error Indicators	118
9.2	Towards <i>a posteriori</i> AMR of the extruded Imperial Front Wing	119

10 Towards h-r-p adaptation	124
10.1 Governing Equations	124
10.2 Shock Capturing and Smoothness Sensor	125
10.3 R-adaptation	126
10.4 P-adaptation	127
10.5 From h to hr - Supersonic intake	129
10.5.1 Uniform Refinement and Reference Solution	129
10.5.2 h-adaptation	131
10.5.3 r and rp-adaptation	133
10.5.4 hr-adaptation	134
10.5.5 Shock Capturing and Accuracy	135
10.6 Full hrp-AMR - Transonic NACA0012	137
10.6.1 Initial Mesh, Uniform Refinement and Reference Solution	137
10.6.2 h-adaptation	139
10.6.3 hp-adaptation	141
10.6.4 hrp-adaptation	142
10.6.5 Solution Accuracy	143
11 Conclusions and Future Work	146
11.1 Conclusions	146
11.2 Current & Future Work	148

List of Tables

3.1	The final association between BRep and mesh objects.	42
4.1	Statistics on the BRep of the Imperial Front Wing.	56
4.2	Comparison of high-order meshes of different types for the eIFW	57
4.3	Statistics on CAD-Reconstruction module. Vertices, Edges and Faces are only counted on the IFW.	61
5.1	Summary of the HO-mesh before and after the isoparametric split. The invalid elements of the initial HO-mesh were linearised before the split.	76
7.1	Capabilities of AMR state-of-the-art simulation frameworks	100
9.1	Overview of the AMR in terms of mesh size, mesh increase and an estimate for the computational cost per Δt_i (last 4 CTU) on HX1 8×64 Intel-Xeon P8358 CPUs.	121

List of Figures

2.1	The isoparametric element notation used in defining the various mappings between the <i>physical</i> , <i>reference</i> , and <i>ideal</i> elements.[104]	26
2.2	Valid straight-sided elements $J_{sc} = 1$ (a), distorted, but valid curved elements $J_{sc} < 1$ (b), an invalid triangular and quadrilateral element $J_{sc} < 0$ (c). Please note the self-intersecting viscous quadrilateral mesh.	28
2.3	Pressure fluctuation (fish-net pattern) occurring in the instantaneous field of an extruded wing due to low mesh geometric accuracy of the underlying CAD (right).	30
2.4	High-order surface mesh of the same case, when the vertex δ is amplified further, showing a spike like pattern due to discrepancy between the straight-sided mesh and the CAD.	30
2.5	A leading edge of an industrial wing cross-section with a curvature radius discontinuity between B-splines on the left. Intentionally coarsened isoparametric edge showing oscillation due to inaccurate mesh curving and same coarse element at $p=5$ curved in the CAD parametric space.	31
2.6	Instantaneous C_p -distributions on a transonic NACA0012 at $M=0.8$ and $AoA=1.25^\circ$ simulated with the same DG-Euler solver at $p=4$ and global de-aliasing (N10) and settings. While P1 meshes are completely inadequate, P2 meshes also show oscillatory behavior. $p=4$ meshes come very close to the expected C_p -distribution on the LE.	32
2.7	A full assembly of an open-source Formula 1 car by Perrinn that consists of 25 subassemblies. [85].	33
2.8	An intersection of two subassemblies of the Perrin F1 car (front wing and nosecone) and a valid geometry with naturally occurring CAD slivers on the endplate of the Imperial Front Wing.	33

2.9	Front wing strakes and their connectivity to the Perrin front wing mainplane with a small angle in the junction, where generating fully hexahedral elements automatically is challenging and the multiple regions of proximity plus cotangency on the HLPW5 extended flaps configuration.	34
2.10	An example industrial CFD, where multiple tools with different fidelity and application targets are connected through different meshing tools and solvers. Note the modularity of the high-order workflow provided as a high-fidelity alternative to current solvers.	35
3.1	NekMesh modular architecture with Input, Process and Output modules	37
3.2	Comparison of the internal bottom-up workflow and the third-party industrial pipeline workflow. Note the lack of reuse of the modules between them.	38
3.3	Mesh distortions due to curving only the edges (p=6) (a) and edge distortion around multi-patch region (p=4) of the legacy pipeline (b).	39
3.4	The new industrial mesh generation workflow starts from third-party linear meshes (blue) which are processed by in-house <i>a posteriori</i> high-order modules (orange).	40
3.5	Overview of the end-to-end mesh generation pipeline, showing all steps/modules in generating the valid high-order mesh. Processes marked in blue are third-party software, whereas all orange ones are internal to NekMesh.	41
3.6	The CAD BRep without the domain a). And a cut through view of the straight-sided mesh of the NACA0012 wing b). It displays the surface triangulation (grey), the boundary-layer prisms (blue) and the tetrahedra outside the boundary layer (brown).	42
3.7	Diagram showing the process of vertex association to BRep objects before projection. Note that none of the vertices is exactly on the BRep. Vertices V_1 and V_3 will have only one associated CAD curve, whereas V_2 will have two.	44
3.8	Diagram showing the 3 different element cases: a) internal face, b) corner (Case 2), and c) multi-patch (Case 3) with initial relaxed and consequently strict association tolerances.	45
3.9	Specific treatment of the objects with multiple CAD surface candidates in a region of trailing edge or corner elements (Case 2).	46
3.10	The 3D parametric projection of the high-order face nodes in parametric (u,v) to physical space for triangular p=4 and quadrilateral p3 faces).	47
3.11	Diagram showing the 3D projection of edge nodes from Cartesian space (x, y, z) onto the closest CAD surface.	48

3.12	The curved surface mesh generated with (b) and without (a) the optimisation. Note the presence of invalid elements when the optimisation is not applied. . . .	49
3.13	Diagram showing the mechanism of assessing the geometrical accuracy $\delta \mathbf{x}_i$ of a $p = 2$ mesh evaluated at $p = 6$ with an evenly distributed quadrature rule. . . .	50
3.14	Measures of geometrical accuracy ($\delta_{\max}$) for the NACA0012 displayed on the linear (left) and the curvilinear (right) surface meshes. Note the different scales of the top (linear) and bottom (curvilinear) legends.	50
3.15	The mappings ϕ from standard reference element Ω_{st} to ideal straight-sided Ω_I^e and the final curved element Ω^e . Note the mapping used for optimization is $\phi = \phi_M \circ \phi_I^{-1}$ [73].	51
3.16	Isoparametric boundary layer splitting in the reference space (ξ) and the mappings from reference to Cartesian space [73].	52
4.1	The high-order mesh generation of a domain around a hemisphere: (a) Initial linear mesh , and (b) the curved surface mesh optimised at polynomial order $p = 9$	53
4.2	Geometrical error of the CAD model (a) and the $p=9$ HO-mesh (b) against the analytical hemisphere showing comparable max errors at the base and the apex with slightly increased δ around the mesh edges. And the geometrical accuracy of the same $p=9$ mesh against the CAD (c), evaluated at evenly spaced points on the HO-mesh, displaying δ_{max} against the CAD being three times lower than the CAD model approximation error.	54
4.3	The high-order mesh generation of a domain around a semi-sphere starting from a linear mesh (a) and the curved surface mesh (b).	55
4.4	The mesh quality of the high-order mesh generation of a domain around a semi-sphere: (a) before and (b) after variational optimisation.	55
4.5	The CAD BRep geometry of the Imperial Front Wing (IFW).	56
4.6	High-order mesh quality distribution for the final optimized four variants of the eIFW, evaluated through the scaled Jacobian.	58
4.7	The 3D linear/initial (left) and high-order $p = 4$ surface mesh (right) of a fully tetrahedral, hexahedral and hybrid prism-tetrahedral meshes, respectively. Last a $p = 3$ invalid ($J_{sc} < 0$) high-order Gmsh mesh improved with the proposed NekMesh pipeline.	59
4.8	A linear mesh of the Imperial Front Wing. The surface mesh is coloured in grey, the prism layers in blue and the tetrahedra in yellow.	60
4.9	Surface quality shown for an enlargement of the isoparametric surface mesh. . .	61

4.10	The maximum elemental geometrical deviation from the BRep of the isoparametric mesh as evaluated at evenly spaced intervals with $p = 6$. Only surface elements with $\delta > 10^{-9}$ are visualised.	62
4.11	Histogram of the distribution of scaled Jacobians in a high-order IFW mesh with $p = 5$ after mesh curving and the end of the meshing workflow.	63
4.12	Time-averaged $\lambda - 2$ vortical structures on the solution at $Re = 220 \times 10^3$ with the Nektar++ incompressible solver using a mixed formulation with $p = 3$ for pressure and $p = 4$ for velocity. [52]	64
4.13	The top of the IFW end plate where a CAD sliver has generated linear elements with high skewness (a), which are easy to tangle during mesh curving or volume optimization. The use of multi-patch elements (b) in the region eliminates the skewed elements.	65
4.14	The mesh generation workflow in reality for more complex than the IFW geometries. In red the new manual interaction in the feedback loop.	65
4.15	An automotive example, where an initially coarse linear (a) mesh resulted in inaccurate mesh curving with tangled elements (b). Final valid mesh after multiple remeshing attempts in StarCCM+ and applying our pipeline (c). Please note the increased curvature refinement and extra DoFs on the other leading edges, where the initial mesh was actually of acceptable mesh quality even with the initial linear mesh.	66
4.16	The IFW cotangent region on the footplate with a fully prismatic macro layer, where the cotangencies have forced the generation of low quality mesh slivers, which are further deteriorating after the mesh curving (red).	67
5.1	High-order refinement, where the split happens with a predefined map in reference space.	69
5.2	High-order refinement, a small part of the >130 alternative maps in reference space to ensure conformity between split.	69
5.3	A coarse Linear Mesh for the Perrin Front Wing	71
5.4	A comparison of a coarse region between straight sided and high-order $p=4$ surface mesh of the Perrinn Front Wing	71
5.5	High-order hp-mesh refinement fully performed in NekMesh for mesh generation. Note that the global resolution of the mesh is preserved and only the distorted elements on the leading edges are refined.	72
5.6	Extraction of the highly distorted elements $J_{sc} < 0.6$ for refinement, before and after proposed process of hp-refinement.	73

5.7	Some of the Additional Prism and Pyramid Maps added	74
5.8	High-order refinement combinations used for boundary layer mesh generation . .	74
5.9	Linear mesh on the left and the high-order surface mesh on the left showing the complexity of the high-lift flap-slat configuration.	75
5.10	Linear mesh on the left and the high-order surface mesh on the left. Note the cotangencies on the slat-mainplane regions.	75
5.11	HO Mesh Pyramids regions corresponding to regions of cotangent surfaces before and after the adaptation.	76
5.12	Mesh quality in terms of J_{sc} for the final HO-split mesh.	76
5.13	Multi-normal region with 5 surfaces having opposing normals making it unsuitable for O-type of prism layer generation. Hence a pyramid region - surface mesh (a) and after the splitting (b).	77
7.1	λ_2 iso-contours coloured by pressure iLES of the IFW by Buscariolo et al. [11] .	86
7.2	Density gradient of a high-pressure turbine stage [81].	87
7.3	Control loop of mesh adaptation	88
7.4	Comparison of the solutions between the CG and DG discretization. Please note the C^0 continuity of the CG formulation and the jump in the solution for the DG.[26]	89
7.5	Adapted mesh ONERA M6 wing: [L^2 -norm (left), adjoint (right)] [4].	92
7.6	r-adaptation applied to resolve a shock [26]	95
7.7	Non-conforming mesh with three hanging nodes.	96
7.8	3D blast simulation, uniform mesh (left), adapted anisotropic mesh (right). [57]	97
8.1	Mechanism behind the isoparametric splitting, from the coarse curved element (1:4 reference map) to four curved and valid child elements. First, the curved parent element transformed to standard space ξ_1, ξ_2 using map ϕ_1 , then a pre-defined map is selected based on which edges are split. This gives the vertices, edges and high-order nodes of the child elements in standard space of the parent element. Finally, one can map the nodes location in physical space and construct the child mesh objects using the ϕ^{-1} also in physical space.	105
8.2	High-Order uniform refinement (1:8 tetrahedral and 1:4 prismatic reference maps) showing transformation from the coarse parent element to the split child elements.	106

8.3	Some of the 130 alternative maps implemented in our high-order h-adaptation. Please note, symmetries or alternatives were used in the hope of avoiding re-ordering tetrahedra for the Nektar++ collapsed coordinate system.	107
8.4	Highly deformed region after isoparametric splitting with initial vertex valence of 8 and the same region after isoparametric swapping in three element pairs resulting in vertex valence of 5 and improved mesh quality.	108
8.5	The effect of mesh smoothing in a region after mesh swapping resulting in an improved overall volume growth rate between elements.	109
8.6	Uniform Isoparametric Refinement of a highly distorted, but valid, initial $p=4$ mesh. Please note the improvement of the minimum J_{sc} from 0.06 to 0.17 and finally to 0.67 for the hhhh-adapted mesh.	110
8.7	Worst case serial performance for uniform h-adaptation for different polynomial orders. Please note the timings do not account for the I/O time of NekMesh. .	112
8.8	Conformal h-adaptation targeting a thin advected front in the semicircle geometry. The process consists of four level of split-swap-smooth calls on the base mesh (a) for every one of the advection stages (b),(c) and (d). Please note the conformity is preserved and mesh valence has not drastically deteriorated in every stage of the process.	113
8.9	Uniform Isoparametric Refinement of a semisphere mesh. Please note only the curved elements of the mesh are presented for clarity purposes.	114
8.10	Uniform Isoparametric Refinement of the eIFW tetrahedral mesh. No CAD-operations were used and the curvature of the base mesh was propagated through the high-order refinements.	114
8.11	Three level h-adaptation in 3D. Note the conformal mesh propagation with alternative maps and the high valence nodes due to this propagation.	115
9.1	Numerical setup of the extruded IFW for the Nektar++ Incompressible N-S Solver for this thesis and [80, 111]	117
9.2	A region marked for adaptation based on the time-averaged over 4 CTUs total pressure (P_0) as the feature-based indicator. Note that the downstream refinement spans from the body to the end of the domain.	118
9.3	Elements marked for mesh adaptation based on time-averaged over 4 CTUs Q -criterion (> 5) and λ_2 -criterion (> 100).	119

9.4	The high-order surface mesh of the initial, h-adapted and hh-adapted hybrid meshes based on the time-averaged Q-criterion. Please note the uniform refinement on the suction side of the wings and the lack of refinement on the pressure side of the mainplane and the first flap.	120
9.5	The high-order surface mesh showing the hhh-adapted mesh with manually selected region for refinement around and downstream the LSB.	121
9.6	Time-averaged pressure coefficient (C_p) distributions on a mid-plane slice of the extruded Imperial Front Wing, comparing the initial mesh (red) and hhh-refined mesh (yellow) matching the same numerical setup with $\Delta t = 1 \times 10^{-4}$ s by Wustenberg (green), but manual remeshing (1.1×10^6 elements).	122
9.7	Time-averaged skin friction coefficient (C_f) distributions on slices of the extruded Imperial Front Wing, comparing the hhh-refined mesh (black) with the same numerical setup with $\Delta t = 1 \times 10^{-4}$ s by Wustenberg, but uniformly remeshed with 1.1M elements. Note the good agreement up to the reattachment region, and downstream up to the end of the manual refinement zone of the hhh-adapted mesh.	123
10.1	Comparison between the initial p=4 mesh used to start all adaptations (a) and four level uniformly refined mesh that is used as a reference "reference" solution (b)	129
10.2	Supersonic Intake at Mach 3 for the reference u_e . Note, inside the intake a strong and weak shocks develop, reflect and propagate, both reaching the outlet.	130
10.3	Persson-Peraire shock sensor. Note the strong and weak shocks develop, reflect and propagate, inside the intake both reaching the outlet. PP illustrated $[-1; -3]$	130
10.4	Mesh and the PP indicator through the AMR evolution. Please note the indicator is limited to $[-3, -1]$ for approximate visualization purposes of what elements are marked for refinement.	132
10.5	Mesh and PP error indicator for the rr-adapted mesh in the study by Marcon [61]	133
10.6	Mesh and PP-sensor for the h-adapted ($h_{min} = 4 \times 10^{-3}$) and the hr-adapted mesh and the PP-sensor for the solutions. Please note the highly resolved strong shocks and the improved weaker ones up to the fourth reflection of the weaker shock.	134
10.7	Comparison between the reference solution (Mach Number) with the h, r [61] and hr solutions on the full top wall of the intake and zoomed in the weaker shocks two and three. Please note that all shocks are captured with both h and hr-AMR, but not with r-adaptation.	136

10.8	Initial mesh with $N_{el}=858$ and four times uniformly refined reference mesh with $N_{el}=318 \times 10^3$	138
10.9	Reference solution, Mach number with overlaid contours showing continous flow without artificial boundary layer created on the slip boundary or artificial dissipation added on the LE.	138
10.10	h-adaptation - mesh and PP-sensor solution evolution. Please note the indicator is limited to $[-3,-1.1]$ for approximate visualization purposes of what elements are proposed to be split.	140
10.11	Final h-adaptation - mesh and PP-sensor solution evolution. Please note the high PP-sensor on the LE and that the indicator is limited to $[-3,-1.1]$ for consistency.	141
10.12	hp-adaptation - mesh and PP-sensor solution. Please note the indicator is limited to $[-3,-1.1]$ for consistency.	142
10.13	hrp-adaptation - mesh and PP-sensor solution. Please note the indicator is limited to $[-3,-1.1]$ for consistency.	143
10.14	Comparison between the different AMR final solutions showing smooth streamlines and sufficient resolution inside the domain. Please note the artificial deceleration around the airfoil for the h-adaptation, starting from the LE. The hp and hrp, on the other hand, show straight Mach profiles starting from the body, as expected for the slip boundary condition.	144
10.15	Comparison of the reference solution with the h, hp and hrp Mach number on the airfoil. Please note the almost full overlap between the hrp and the reference solution and the effect of LE artificial dissipation for the h-adaptation, offsetting both the acceleration region and the recovery region downstream the weaker shock.	145
11.1	h-adaptation on NACA0012 wingtip on vortical dominated flow driven by Q-criterion and the improvement in vortical resolution on the wing tip trailing edge. [46]	149

Chapter 1

Introduction

Computational Fluid Dynamics (CFD) has become the status quo for designing, verifying and even validating complex industrial products in areas such as aerospace, automotive, turbomachinery and many others. These industries heavily employ CFD simulations to refine their designs, optimize component performance, and shorten development cycles slowly but surely decreasing the time and cost spent for large-scale experimental testing campaigns.

For decades, low-fidelity finite volume methods (FVM) solving steady Reynolds-Averaged Navier–Stokes (RANS) equations have dominated the field. They have served as the primary industrial design tool, thanks to their low computational cost, stability, strong track record, and extensive industrial experience in all kinds of engineering applications with final designs typically validated against more expensive windtunnel experiments. The ongoing improvements in computer hardware and the growing need to capture inherently unsteady behaviour in many real-world applications are driving the adoption of hybrid methods such as Detached Eddy (DES), Large Eddy Simulation (LES) or even Direct Numerical Simulations(DNS). Although the latest studies have been limited only to simple academic cases, they are now rapidly approaching everyday industrial use cases. Precisely for these high-fidelity LES/DNS applications, academia has demonstrated a compelling and in many respects, natural evolution of the low-order finite volume methods - high-order ones, such as Spectral/hp Element (SEM) and Discontinuous Galerkin(DGSEM) methods. The higher-order polynomial expansions representing the solution within each element have been shown to keep the flexibility of FVM methods, but also provide greater accuracy for a given mesh resolution and a potentially superior dissipation and dispersion behaviour. When all these properties are combined with modern high-performance computing advances such as GPUs, the intrinsic high arithmetic intensity and algorithmic efficiency of local matrix free operators, high-order methods can offer detailed insights into unsteady flow physics based on approaches like LES, while maintaining competitive computational efficiency and industrially viable cost. This makes them one of the prime targets

for any industry focused on cost and efficiency.

However, realising the full potential of these methods requires coarse high-order meshes that accurately represent complex industrial geometries while maintaining good mesh quality for both the surface and the internal mesh volume, even when highly stretched curved elements are required. To date, a major bottleneck in the industrialisation of high-order SEM remains the automatic and robust generation of high-quality, body-fitted viscous curvilinear meshes for very complex geometries without simplifications or the need for manual intervention on the CAD geometry or manual iterative mesh modifications.

Moreover, the increased geometrical complexity and growing interest in high-fidelity simulations, such as LES, have forced researchers to address the ever-growing computational demand for locally refined meshes. These meshes are essential for capturing small-scale flow structures with their corresponding mesh resolution requirements of $(\Delta x^+ \Delta y^+, \Delta z^+)$. As this can be done only a-posteriori once a solution is present, adaptive mesh refinement (AMR) strategies are also of interest to this work. High-order methods provide an even more sophisticated trade-off for AMR, where accuracy and cost can be controlled by both mesh resolution h and the local polynomial order p . This unique *a posteriori* hp-adaptation capability is still an open research question even for simpler geometries and could open the door to ever more accurate, efficient, and sustainable CFD in both industry and academia. These two general topics, along with the potential overlap between them, will be the focus of this thesis.

Thesis Outline

- I. High-Order Mesh Generation for Complex Geometries from third-party meshes and its integration within already existing industrial workflows.
- II. *A-posteriori* High-Order Mesh Adaptation for complex geometries in the context of compressible and incompressible flows.

Publications

Part I

- **K.S.Kirilov**, J. Zhou, M.Green, J. Peiró and D. Moxey, High-Order Curvilinear Mesh Generation From Third-Party Meshes, Proceedings of the 2024 International Meshing Roundtable (IMR), 2024, doi 10.1137/1.9781611978001.8
- M. Green, **K.S.Kirilov**, et all NekMesh: An Open-Source High-Order Mesh Generation Framework, Computer Physics Communications, 2024. doi 10.1016/j.cpc.2024.109089
- **K.S.Kirilov**, J. Zhou, J. Peiró and D. Moxey. High-Order Curvilinear Mesh Generation From Third-Party Meshes, Special Issue of Computer-Aided Design Journal, 2026. doi 10.1016/j.cad.2025.103962
- P. Khurana, A. Liosi, **K.S.Kirilov** et all. Industrialisation of spectral/hp element method for incompressible, transitional flow around Formula 1 geometries, Journal of Wind Engineering and Industrial Aerodynamics, 2026 (Manuscript in Preparation)

Part II

- **K.S.Kirilov** et all., h-r-p Adaptation for Compressible Flows, Computer Methods in Applied Mechanics and Engineering (Manuscript in Preparation)

Other

- **K.S.Kirilov**, J.Peiró, et all, Curvilinear Mesh Generation for the High-Order Virtual Element Method (VEM), 2024 Proceedings of SIAM International Meshing Roundtable 2023. Lecture Notes in Computational Science and Engineering, 978-3-031-40594-5_19
- **K.S.Kirilov**, J.Zhou, J.Peiró, et all, Mesh generation of curvilinear polygons for the high-order virtual element method (VEM), Special Issue of Computer-Aided Design Journal, 2026. doi 10.1016/j.cad.2025.103966

Part I

High-Order Mesh Generation

Chapter 2

Literature Study & Research Problem

2.1 The High-Order Element

High-order methods are capable of encoding more information about the solution by the combination of high-order basis functions l_i in a single element compared to their low order counterparts [42]. Hence, they tend to be significantly larger in size for an optimal performance. This, combined with the lower diffusion characteristics of the methods require the elements themselves to be curved to the desired underlying geometry or along the flow features. While there have been different methodologies to describe their solution and the shape by different basis functions such as NURBS basis functions for the Isogeometric analysis [38], this thesis will focus only on standard spectral/hp element method (SEM) as described by Karniadakis & Sherwin [42], where the mesh is constructed by conformally connected unstructured Isoparametric elements.

The **isoparametric element** will refer to an element where the physical geometry and solution variables are both described using the same set of polynomial shape functions l_i inside the reference space ξ . This way the problem of solving a mathematical operation is transferred from Cartesian space $\mathbf{x}$ to local reference space ξ , where the element is always described locally in $[-1,1]$. From a geometrical perspective this allows to describe a very complex curved element using this isoparametric map ϕ_p from the reference element to the final curved element in Cartesian space, see Figure 2.1. This approach ensures that the underlying curved geometry can be approximated accurately. This is particularly important for spectral/hp element methods, which are known to for their accuracy and higher sensitivity to small perturbations. Such geometrical perturbations can potentially degrade the accuracy and stability of the simulation, making a precise curved geometrical representation essential.

In SEM the solution field within each element is approximated using a tensor-product expansion

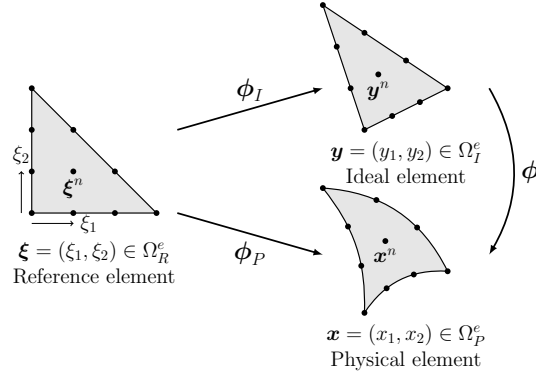


Figure 2.1: The isoparametric element notation used in defining the various mappings between the *physical*, *reference*, and *ideal* elements.[104]

of Lagrange polynomials defined on the reference element. Specifically, for a two-dimensional quadrilateral element, the solution u is represented as:

$$u(\xi_1, \xi_2) = \sum_{i=0}^N \sum_{j=0}^N u_{ij} \ell_i(\xi_1) \ell_j(\xi_2), \quad (2.1)$$

where $\ell_i(\xi)$ are the Lagrange basis functions associated with Gauss-Lobatto-Legendre (GLL) nodes. Similarly, the geometry of the element can be described using the same basis functions, ensuring an isoparametric formulation:

$$\mathbf{x}(\xi_1, \xi_2) = \sum_{i=0}^N \sum_{j=0}^N \mathbf{x}_{ij} \ell_i(\xi_1) \ell_j(\xi_2), \quad (2.2)$$

where $\mathbf{x}_{ij} = (x_{ij}, y_{ij})$ are the physical coordinates of the element nodes.

The transformation from the reference element to the physical element involves the Jacobian matrix of the mapping $\mathbf{J}$ which is essential for integrating over the physical element and the scalar determinant of this mapping J , respectively given by:

$$\mathbf{J}(\xi_1, \xi_2) = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}, \quad J(\xi_1, \xi_2) = \det \mathbf{J}(\xi_1, \xi_2), \quad (2.3)$$

Alternatively, for the triangular elements, the SEM employs a hierarchical or modal basis, leading to the following approximation for the solution field:

$$u(\xi_1, \xi_2) = \sum_{i=0}^N \sum_{j=0}^{N-i} \hat{u}_{ij} \phi_{ij}(\xi_1, \xi_2), \quad (2.4)$$

where $\phi_{ij}(\xi_1, \xi_2)$ are orthogonal basis functions defined on the reference triangle.

Finally, the numerical integration over the curvilinear element becomes:

$$\int_{\Omega^e} f(\mathbf{x}) d\mathbf{x} = \int_{\Omega_{st}} f(\mathbf{x}(\boldsymbol{\xi})) J(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (2.5)$$

2.2 High-Order Mesh Generation and its Challenges

The most common way to generate a high-order mesh is to first create a low-order straight-sided mesh through a classical finite volume approaches and then curve it to the underlying geometry, which is best known as *a posteriori* high-order mesh generation. While there has been a recent surge in direct meshing approaches such as Bezier Guarding[44] or direct NURBS enhanced discretization as in NEFEM [113], this thesis will focus only on high-order *a posteriori* approaches. This choice usually encounters certain challenges during the mesh curving and propagation of curvature for highly-stretched elements to capture boundary layers at high Reynolds Numbers (Re). Namely *Self Intersections* and *Mesh Quality, Geometrical Accuracy* and *Ensuring Desired Mesh Resolution*. These have been targeted by multiple research groups in the past 10-15 years and some maturity has been achieved [104], [32], [34], [90]. However, the introduction of these processes into an industrial environment with complex 3D geometries creates some additional challenges that have not yet been widely addressed - namely: *CAD Quality and Geometrical Complexity* and the desire for maximum *Integration* with already existing low-order CFD pipelines. To investigate these problems, determine their severity for industrial applications and the current state of the art, a research study was performed both based on literature, personal contributions and numerous collaborations within the SSECOID project, the Nektar++ group and industrial partners.

2.2.1 Mesh Quality for Coarse Highly Stretched Viscous Layer Meshes

The problem of mesh quality is probably the one that has received the most attention in the past decades, both from evaluation and from mesh optimization perspectives. There have been numerous ways to define the mesh quality of a straight-sided and curved elements by Branets [9], Roca [92] and particularly the work by Knupp [48], where the element is described in terms of shape, size and orientation, which is actively extended to multi-metric control by Dobrev and Tomov [107], [24] to name a few. While all of these have different approaches in defining quality and "the best mesh" *a priori*, they all acknowledge the validity region of a high-order curved element is a function of the deformation of the element from its reference and ideal elements.

Highly distorted elements, see Figure 2.2 b) have multiple detrimental effects on the solution of a high-order solver. First and foremost they tend to have highly irregular mapping ϕ between

the reference and physical space. This leads to increased numerical errors of any operator performed in the reference space of the element, for example integration, differentiation, etc. This will naturally impact negatively the convergence and the overall accuracy of the simulation. Moreover, experience has shown a significant deterioration of the conditioning of the linear system and hence lower stability in presence of highly deformed elements. Moreover, the two negative effects can become synergetic - the numerical errors generated by the operators will be the most amplified in the most deformed parts of the elements and could restrict the stability margin through the effective Courant–Friedrichs–Lewy (CFL), being influenced by the shortest distance between quadrature nodes as proposed by Jorshi and Ferrer [41].

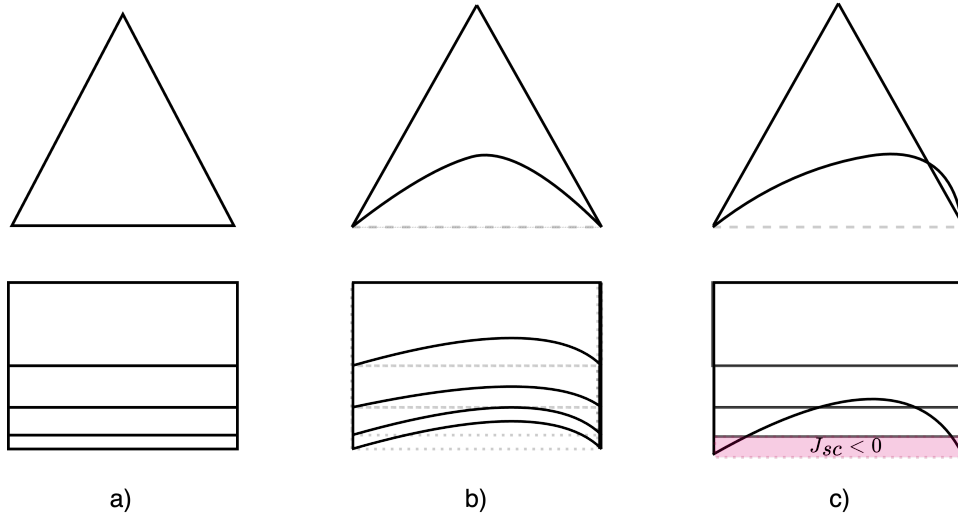


Figure 2.2: Valid straight-sided elements $J_{sc} = 1$ (a), distorted, but valid curved elements $J_{sc} < 1$ (b), an invalid triangular and quadrilateral element $J_{sc} < 0$ (c). Please note the self-intersecting viscous quadrilateral mesh.

Deforming the element further will lead to a self-intersection, see Figure 2.2 c). Mathematically this will invalidate one of the main requirements for high-order methods using Galerkin projection - to be orientation preserving of ϕ and hence might diverge any simulation. Hence, the formal validity of a mesh can be defined as a positivity of the Jacobian defined continuously. However, for any practical purposes the evaluation is done in the selected quadrature points of the element and the validity is defined as strictly positive Jacobian in all evaluated quadrature points. As this value varies across the curved element, a common way to describe the elemental quality is the scaled Jacobian, see Equation 2.7 with optimal value of $J_s = 1$ corresponding to a straight-sided element and negative for invalid with self intersection.

$$J(\xi)_s = \det |\nabla \phi_M| \quad J(\xi)_s > 0 \quad (2.6)$$

$$J_s^e = \frac{\min J(\xi)}{\max J(\xi)} \in [-\infty : 1] \quad (2.7)$$

When dealing with high Reynolds-number flows, the boundary layer can be extremely thin and sensitive to mesh resolution. Achieving robust, body-fitted meshes with highly stretched elements (large aspect ratios) in the wall-normal direction is essential to capture viscous phenomena. Unfortunately, this results in highly deformed and invalid elements that are almost impossible to avoid. In these cases the initial straight-sided mesh is incapable of accommodating the mesh curving if a standard interpolation on the CAD is performed, see Figure 2.2 c).

Therefore, there has been multiple approaches for mesh curving a straight-sided mesh or optimizing an invalid low-quality high-order mesh for untangling: starting from Persson and Peraire who laid the foundational techniques for mesh curving through mesh deformation using solid mechanics principles [87]. Another noticeable series of work by the Barcelona Supercomputing Group lead by Roca, who contributed to approximative mesh curving on both simple and industrial geometries [32] with sliding nodes on the CAD BRep [94].

The Target-Matrix Optimization Paradigm (TMOP) by LNLL MFEM group [21] pushed the r-adaptivity and GPU-native mesh optimization for ALE hydrodynamics with quality control through metric-driven approaches defined by Knupp & Tomov [20] with a main goal of never tangling an already valid mesh.

These approaches have found their way to commercial software like Pointwise through the work by Karman who developed weighted condition number (WCN) *a posteriori* mesh curving methods for viscous meshes and with mixed order [96]. It is also important to mention the contributions by Remacle, Geuzaine and the GMSH group using a optimization with moving log-barriers for untangling and curving capabilities [103].

Last, the work by Moxey et al. used a physical thermoelastic analogies for a posteriori mesh curving [72], which was then extended drastically into a variation framework that generalized all PDE approaches by Turner into one optimization framework [106] capable of both curving and untangling already invalid meshes. This variational optimizer was then incorporated into the Nektar++ solver and foundational for the high-order mesh generator NekMesh [35] and this thesis.

2.2.2 Geometrical Accuracy & Sources of Geometrical Perturbations

Previous studies have examined the sensitivity to geometrical accuracy, although not as deep as mesh quality. For example, Nordström [75] and Nelson & Kopriva [50] mainly focused on interpolative mesh curving of academic geometries, while Ruiz-Gironés and Roca [95] introduced geometrical accuracy as an optimization metric for approximative mesh curving. All these

studies indicate that higher geometrical accuracy is needed compared to basic linear, straight-sided meshes. We extend these studies and investigate some of the causes of geometrical errors and the minimum necessary order for the spectral/hp element methods.

Vertex Deviation In the beginning of this project, we applied a legacy mesh curving techniques and simulated simplified 3D industrial geometries within the industrial collaboration. This lead to some unphysical oscillations being present in the instantaneous pressure fields, see Figure 2.3. After detailed investigation, mesh hp and solver p refinement studies, it was concluded that the unexpected behavior persists, and a causation was established to the geometrical accuracy of the underlying mesh, fishnet pattern following the element edges. This problem propagated even more distinctly in the shear stresses.

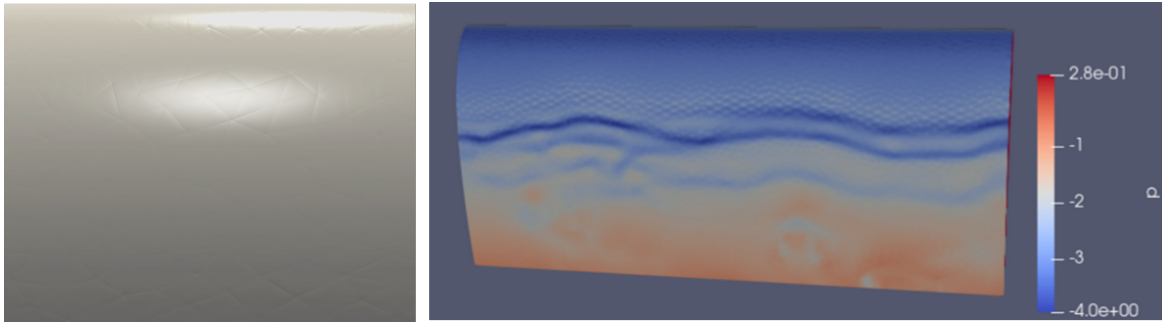


Figure 2.3: Pressure fluctuation (fish-net pattern) occurring in the instantaneous field of an extruded wing due to low mesh geometric accuracy of the underlying CAD (right).

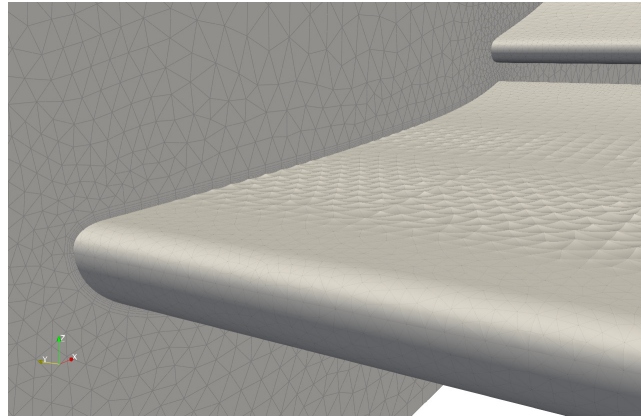


Figure 2.4: High-order surface mesh of the same case, when the vertex δ is amplified further, showing a spike like pattern due to discrepancy between the straight-sided mesh and the CAD.

In this case, a max vertex deviation $\delta_x = 4.6 \times 10^{-4}$ m of the straight-sided finite volume mesh resulted in inaccurate mesh curving and fishnet like mesh artefacts that were visible on the curved isoparametric surface mesh, see Figure 2.3. For clear visual of the surface mesh deterioration, we amplify this discrepancy and see spike-like pattern on the pressure side of this wing, see Figure 2.4. Identical problem has been present even in the finite volume community, as reported for the same geometry by O’Sullivan [79] for the CharLES solver. This underlines the importance of the conformity of the initial straight-sided mesh vertices to the CAD.

Naive Mesh Curving to Closest CAD Another observation from these studies was the necessity for an accurate mesh curving technique for coarse elements. In some of these cases, see Figure 2.5 a) , C^0 , but more visibly C^1 wall normal deviation is present due to a naive mesh curving algorithm (interpolative from the straight-sided quadrature node to the closest CAD location). For coarse meshes of industrial significance, it is possible to observe such mesh distortions with high-polynomials ($p=4-5$) especially when non-uniform curvature geometries are of interest. This requires ensuring the geometrical accuracy of the interpolated mesh, either through approximative mesh curving or parametric projection happening in the CAD parametric space, so both C^0 Hausdorff distance and C^1 normal errors on the quadratures points are acceptable, (see Figure 2.5 b). This is also supported by the work of Ruiz-Gironés and Roca [95], showing some limitations of interpolative mesh curving and even suggesting the use of through approximative meshes.

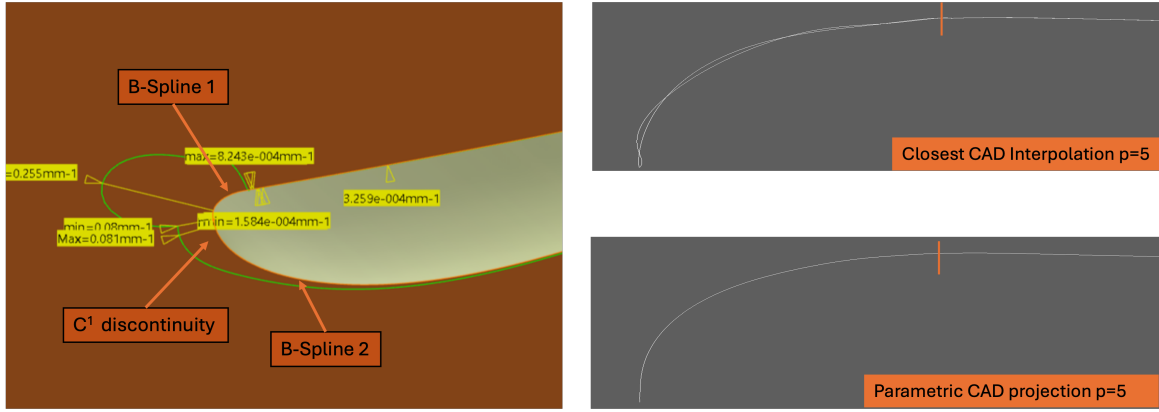


Figure 2.5: A leading edge of an industrial wing cross-section with a curvature radius discontinuity between B-splines on the left. Intentionally coarsened isoparametric edge showing oscillation due to inaccurate mesh curving and same coarse element at $p=5$ curved in the CAD parametric space.

Minimum necessary order The previous problems motivated a numerical study in 2D using the mainplane of Imperial Front Wing (Incompressible N-S Solver) and NACA0012 (N-S Solver) for understanding the required geometrical accuracy and polynomial order of the mesh. For simplicity, the Euler compressible flow solver was used at Mach number $M=0.8$ at $AoA=1.25^\circ$ and the C_p -distributions for an under-resolved mesh were simulated with a mesh curving at polynomial order (p) varying from $p \in [1-5]$. To distinguish the geometrical error from the other numerical errors, exactly the same linear mesh, initialization, solver settings, computational resources and fixed solver polynomial order were used for all simulations combined with minimal numerical dissipation to expose the geometrical effect clearly.

This simple study underscored the critical necessity for generating high-order meshes with mesh curving degree higher than $p = 3$ to resolve reliably the geometry and accurately the flow phenomena without triggering geometrically driven oscillations. For the lower polynomial cases: $p = 1$ was completely inadequate and $p = 2$ while having good C^0 accuracy within the element,

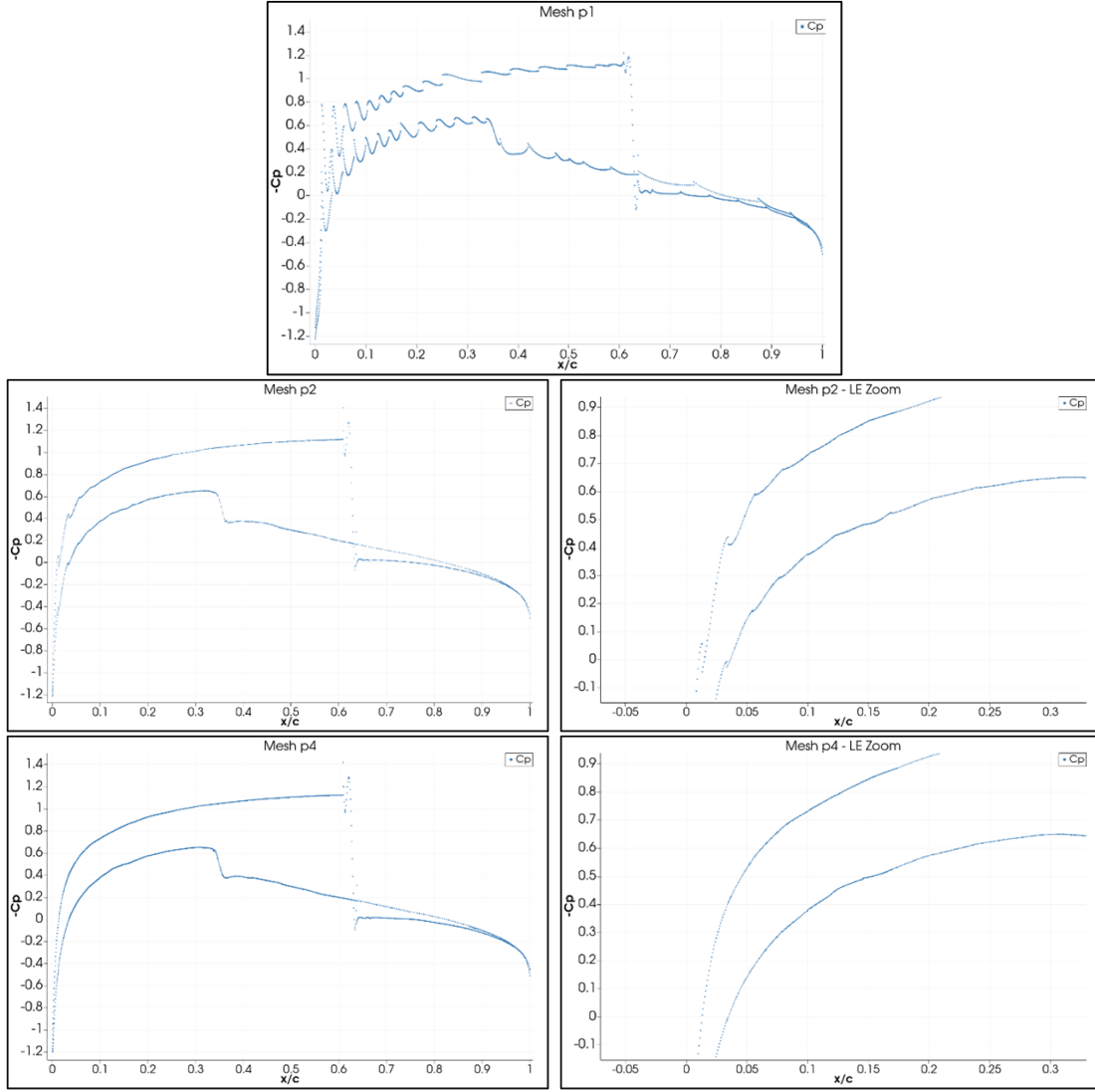


Figure 2.6: Instantaneous C_p -distributions on a transonic NACA0012 at $M=0.8$ and $AoA=1.25^\circ$ simulated with the same DG-Euler solver at $p=4$ and global de-aliasing (N10) and settings. While P1 meshes are completely inadequate, P2 meshes also show oscillatory behavior. $p=4$ meshes come very close to the expected C_p -distribution on the LE.

it exhibited oscillations triggered by the C^1 discontinuities between elements. Alternatively, the geometrical errors can be alleviated by refining the mesh, but this is suboptimal for the computational cost of the simulation and has limited effect on discontinuities between elements. Last, it is worth noting that this is for a simple smooth geometry which can be considered the best case scenario and even higher requirements are necessary for more complex cases.

2.2.3 CAD Quality

Complex Industrial CAD models exhibit intricate curvatures, sharp edges, subtle design features and various imperfections. Even just cleaning them into a valid watertight representation that can be accurately curved for high-order elements, requires experience and could take days and weeks until a linear, straight-sided mesh is generated [93]. In industries where thousands

of parts may be merged into a single assembly, see Figure 2.7, that is governed by CAD kinematics or morphing will always result in minor misalignment, gaps, intersections, non-manifold connectivity and topologically impossible to mesh features.

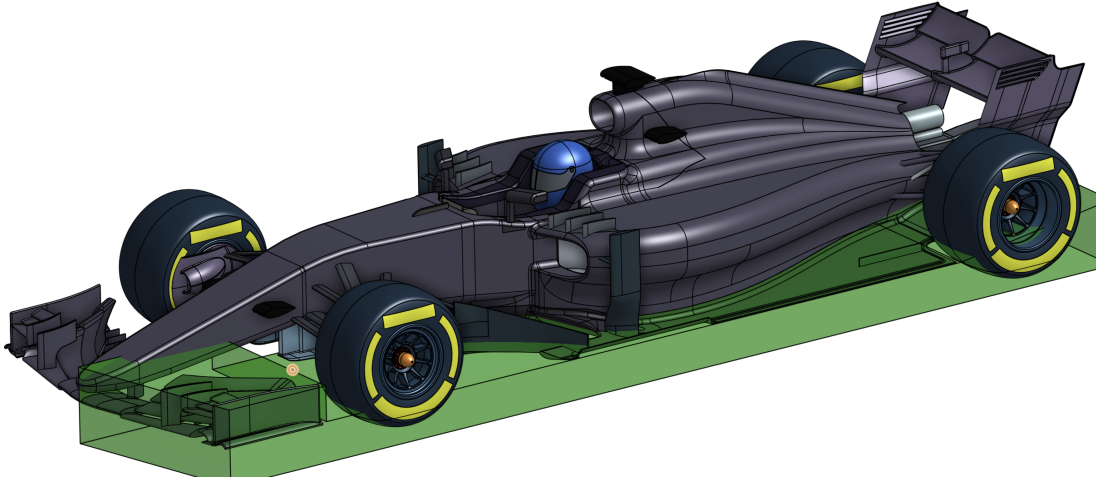


Figure 2.7: A full assembly of an open-source Formula 1 car by Perrinn that consists of 25 subassemblies. [85].

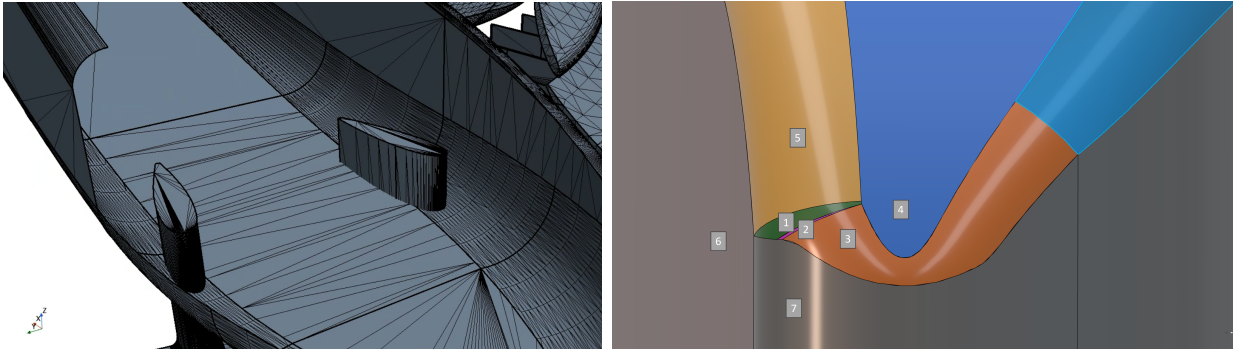


Figure 2.8: An intersection of two subassemblies of the Perrin F1 car (front wing and nosecone) and a valid geometry with naturally occurring CAD slivers on the endplate of the Imperial Front Wing.

In the best-case scenario, like in fast-paced industries like Formula 1, the real CAD might be watertight but still full of incompatible features for direct surface triangulation meshing on the BRep as CAD slivers, overlapping CAD curves and other artefacts, such as very thin solids, etc will always be present. These can be caused by the CAD software itself or by the design operations, such as fillets or joints that are still likely to cause low mesh quality, see Figures 2.8, forcing the tool to rely on virtual geometries spanning over two or more surfaces that might not be even C^0 continuous, but still preserving sharp features.

Hence, even for the surface mesh, one needs to rely on extremely robust linear mesh generators to be able to mesh through these imperfections or be ready to spend days to weeks of CAD repair with CAD software or commercial tools like ITI CADfix [39] or ANSA [8].

2.2.4 Geometrical Complexity and Element Flexibility

A natural choice for a high-order solver is the hexahedral element due to its natural mathematical definition of the tensor product and making it into efficient matrix-free operators with sum-factorization, giving it usually higher throughput in comparison to other types of elements [69] and practical support in all high-order solvers.

However, even in the previously described best-case scenario and with a valid surface mesh, the CAD geometry is still extremely complex, possessing multiple regions of high curvature, thin features as Trailing Edges (TE), peculiar junction between surfaces and regions of proximities of non-connected CAD surfaces, see Figure 2.9 a). And while there has been recent progress in fully quadrilateral surface [15] and fully hexahedral linear volume mesh generation, for example, by Remarcle, Bommers, etc [89] respectively, to the best of the authors' knowledge, their state is far from allowing fully automatic body-fitted meshing of all irregular regions with viscous layers. Therefore, currently one can expect only a hybrid meshes containing all standard types of unstructured elements (prism-pyramid-tetrahedra-hexahedra) for such complex geometries.

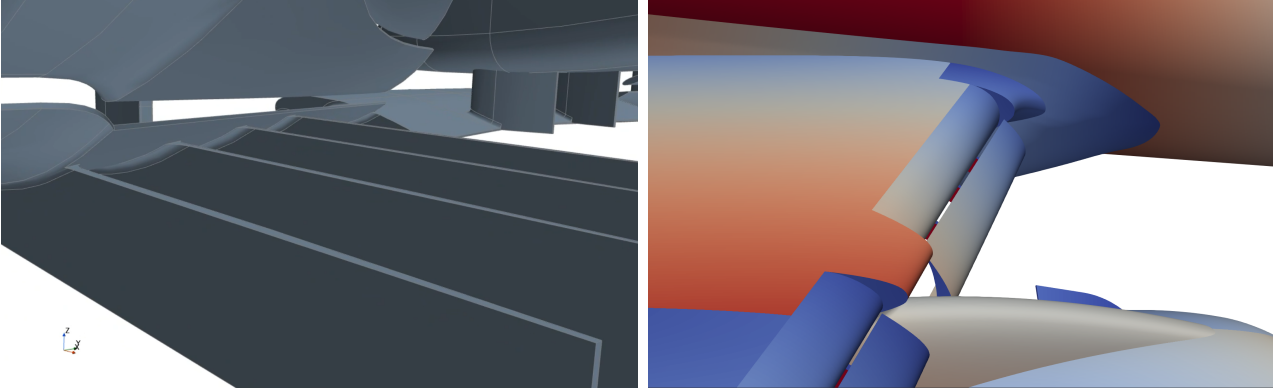


Figure 2.9: Front wing strakes and their connectivity to the Perrin front wing mainplane with a small angle in the junction, where generating fully hexahedral elements automatically is challenging and the multiple regions of proximity plus cotangency on the HLPW5 extended flaps configuration.

An ultimate high-order meshing workflow should be able to handle and curve all type of unstructured elements for the current industrialization, be prepared for a potential breakthrough in fully hexahedral or hex-dominant mesh generators, but be applicable to novel methods where an element generalization is achieved through the definition of polyhedra as the Polyhedral Discontinuous Galerkin Methods or the Virtual Element Method.

2.2.5 Automation and Integration in Existing CFD Processes

Throughout this thesis, there were extensive collaborations with industrial and academic partners, as well as potential end-users. The academic users were open towards creating from

scratch a manual dedicated to a high-order workflow with its specific way for generating the geometry, the mesh and high-order specific post-processing utilities. However, the industrial users heavily favoured integrating the high-order workflows into their existing processes, combining the tools with different pre- and post-processing tools, see Figure 2.10. They usually employ specific meshing tools and solvers for different applications, and have a well-defined process for each of them, sharing bits in between, as for example PADRAM [66] for turbomachinery.

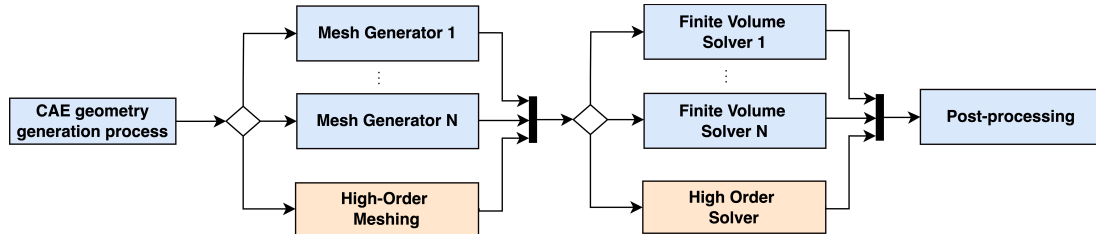


Figure 2.10: An example industrial CFD, where multiple tools with different fidelity and application targets are connected through different meshing tools and solvers. Note the modularity of the high-order workflow provided as a high-fidelity alternative to current solvers.

This is expected as a company has already invested significant effort and resources into their existing CFD process, licenses and has experience in making them robust, troubleshooting and most importantly designing based on their CFD results. Moreover, the demand for high simulation throughput of some industries, like Formula 1 with routinely up to 1000 low fidelity simulations a month, is another incompatible requirement with manual CAD repairing and meshing processes. Although the higher fidelity solvers might not initially take the full load, it is an absolute necessity to facilitate such direct integration with existing processes and full automation with minimal manual interaction from the user. Last, even if physically possible, specific CAD modifications or simplifications for the high-order meshing could not even be allowed at all due to industry regulations, for example, in Formula 1, due to the RATG limitations.

On the high-order meshing side, this means virtually no CAD repair or simplifications, extreme linear mesh generation robustness, once a CAE model is provided. If this is available, a high-order mesh should come out at the end of the process without extensive iterations, even if it is not a perfect one. This requires modularity and potential for integration with different pre-processing CAD input, modification inside the high-order tool, and even curving meshes produced in third-party commercial or open-source meshing software.

2.3 Current State of the Art

At this point, it is possible to analyze the current state-of-the-art for complex geometries, putting it in the context of industry. Starting from the commercial world, one must mention the recent work by Pointwise, currently part of CADENCE. It has been the first and probably

the most versatile high-order mesh generator following their (WCN) *a posteriori* approach [90] and wide finite volume meshing capabilities. It provides a good balance between robustness for unstructured viscous meshes and mesh curving up to $p = 4$. GridPro [2], on the other hand, has focused on block-structured meshes, though only on simpler geometries with manual work. During the duration of this PhD project, ANSA [8] introduced a high-order mesh curving capability. This is promising given ANSA’s robustness for tackling invalid CAD; however, some feedback from its developers suggests the high-order feature is still in its early stages of development. Of course, all this robustness comes at the cost of additional commercial licenses and adoption, which might be prohibitive for many users, especially in companies relying on open-source CFD software—it might well be a limiting factor for high-order methods in CFD.

Since academia has been at the forefront of high-order meshing for over two decades, it is worth mentioning tools like Gmsh, MFEM, NekMesh, but also the BSC-CASE group. Gmsh is currently the most widely adopted open-source mesh generator in general, with relatively good robustness in both high and lower order meshing, but has limited capability for handling non-watertight geometries and rather limited support for automatic viscous layer meshing. The LLNL MFEM tools provide probably the state-of-the-art optimization and adaptation with exascale performance, but with no initial mesh generation. The BSC-CASE group has shown the state-of-the-art in terms of approximate mesh curving with its demonstrations on the High-Lift Prediction Workshop, but with only fully tetrahedral elements and its tools are not openly available. Last, NekMesh, which is also the focus of this thesis, has also shown its capabilities for interpolative mesh curving and untangling for hybrid prismatic-tetrahedral meshes, but the linear viscous mesh generation was not deemed robust enough for very complex geometries, leading to the initial work on third-party high-order meshing by Turner [65] and subsequently this thesis.

2.4 Research Objectives

Given the presented literature study, the current state-of-the-art and the direct exposure to industrial partners, the author aims at contributing to the practical adoption of high-order methods in industry by removing the barriers of high-order mesh generation for very complex geometries. Hence, the main objectives of the thesis are outlined:

1. **Develop a Robust End-to-End High-Order Meshing Workflow**
2. **Ensure High-Quality Viscous Body-Fitted Meshes**
3. **Ensure Element Type Flexibility for Extreme Geometrical Complexity**
4. **Ensure Seamless Integration into Existing CFD Workflows**

Chapter 3

Proposed Methodology

3.1 High-Order Mesh Generation with NekMesh

*If I have seen further, it is by standing
on the shoulders of giants.*

— Isaac Newton, 1675

This thesis is part of NekMesh, the open-source high-order mesh generator of the hp/SEM framework Nektar++. Hence, it is worth providing a brief overview of the tool and its capabilities as they were before this project, giving a further context for some design decisions and philosophies. Moreover, it is important to emphasise that neither the methodologies nor the meshes produced in this work are exclusive only to Nektar++ - they target the broader high-order hp/SEM isoparametric community and are compatible with different solvers, point distributions and basis functions. [19] NekMesh supports both linear and high-order mesh generation, CAD interaction through CAD-APIs of STEP and conversion between various mesh formats. Its modular architecture, organised into Input, Process, and Output, see Figure 3.1, fits directly into the previously described idea for "plug and play" into an industrial pipeline.

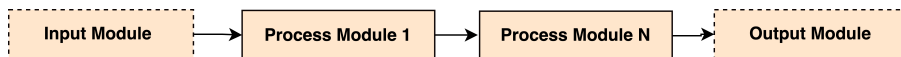


Figure 3.1: NekMesh modular architecture with Input, Process and Output modules

This leads to two main mesh generation workflows. The first one is the **internal bottom-up mesh generation** workflow, where the linear hybrid mesh is generated directly from the BRep by NekMesh itself with a modified version of packages Triangle [98] and TetGen [100]. Then, retaining the full connectivity with the CAD, the mesh is curved through a parametric projection, surface and volume mesh optimization modules [35]. Last, the high-order mesh

is split isoparametrically to ensure the desired y^+ distribution around the body. This Workflow has been shown to produce NekMesh’s highest quality and geometrical accuracy meshes. However, it does require a near-perfect watertight BRep, has limited control and flexibility for the mesh refinement control, and most importantly, lacks robustness for generating stretched viscous meshes in surface proximity regions. This limits its usage to mainly academic cases with relatively medium complexity, but prohibits it for very complex geometries, the problem being the generation of the advancing front prism layer.

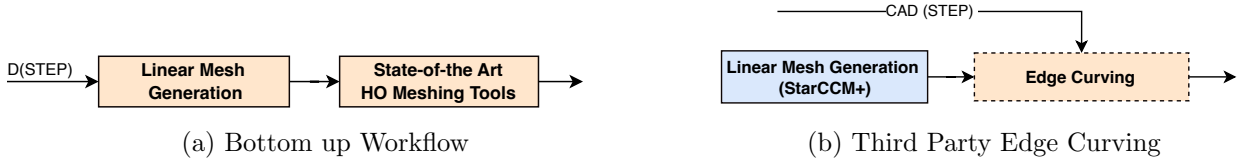


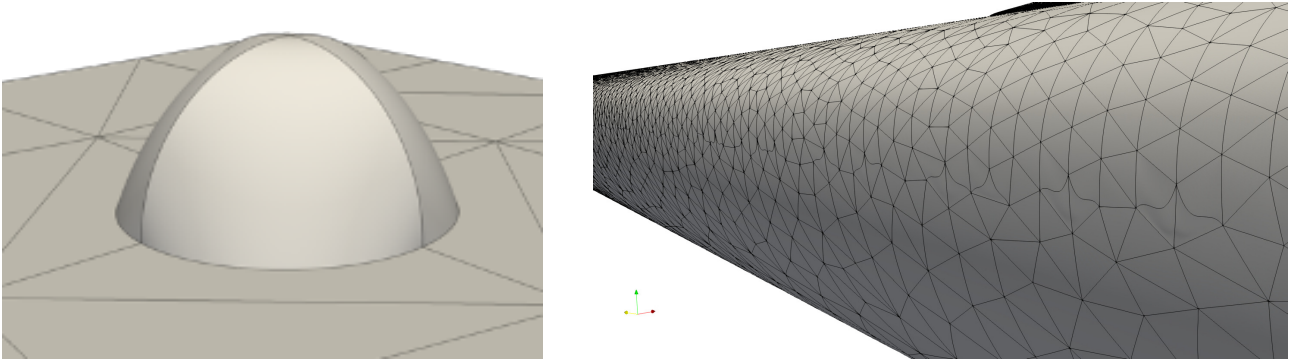
Figure 3.2: Comparison of the internal bottom-up workflow and the third-party industrial pipeline workflow. Note the lack of reuse of the modules between them.

Consequently, as per the BSC group, our effort has been focused on using third-party linear mesh generators, which bring multiple advantages - the previously mentioned robustness and flexibility, but also decades of existing industrial know-how and automation. However, the a posteriori methods often rely on some critical assumptions that are not necessarily satisfied when using third-party linear meshes:

- **CAD-conformity:** The vertices of the linear mesh lie on the CAD curves and surfaces within machine precision (ϵ_M). Such errors can occur due to FVM mesher placing face centroid rather than vertices on the CAD, or due to discrepancy between the original BRep and the CAD-geometry used for the linear mesh generation.
- **CAD-connectivity:** The mesh curving and optimisation procedures require accurate information about the corresponding BRep object.
- **CAD-compliance:** The mesh curving processes may assume that every mesh face corresponds to precisely one BRep surface, which is not always the case for third-party meshes.

The majority of commercial mesh generators do not retain the link between the mesh object and the BRep one, which is necessary for both mesh curving and optimization techniques. The first working attempt to overcome this was implemented by Turner [104], which allowed an increase in the complexity of the curved meshes through simple projections of the edges to the closest CAD surface and CAD non-compliance for the edges. This module proved to be highly effective, enabling some of the most complex high-order simulations at that time, such as the RP1 car [65] and the Imperial Front Wing by Buscariolo [11].

It was specifically implemented for tetrahedral or tet-prismatic meshes, mainly coming from StarCCM+. However, due to the absence of the full CAD-connectivity, a dedicated module was required for analytical edge projection, and the "state-of-the-art" mesh curving and optimization modules could not be directly applied. Consequently, the resulting meshes were often highly refined or lacked geometrical accuracy and had some of the artefacts discussed in Section 2.2. Please, refer to Figure 3.3 for two extreme examples. Moreover, most of the stringent requirements on the CAD BRep validity were similar to the academic workflow, hindering the full automation and consequently requiring an extensive manual CAD-clean up, iterative manual modifications in the third-party mesh generator, see Figure 3.3.



(a) HO-mesh of a semisphere with 4 elements.

(b) Fine HO-mesh with distorted elements at CAD junction.

Figure 3.3: Mesh distortions due to curving only the edges ($p=6$) (a) and edge distortion around multi-patch region ($p=4$) of the legacy pipeline (b).

3.2 Proposed Methodology

In this work, we introduce a generic approach that leverages the strengths of both pipelines. We combine the geometric accuracy and mesh quality coming from the mesh curving and optimization techniques of the academic pipeline with the robustness and flexibility of any third-party linear mesh generators.

The starting point is a coarse, straight-sided (linear) volume mesh in a third-party mesh generator such as TAU meshers [33], StarCCM+ [101], Fluent, ANSA[8], etc. From that point of the workflow, the goal is to produce a valid high-order mesh automatically from any linear mesh without relying on a specific tool, specific mesh type or extensive iterative interventions.

To achieve that, we propose a new intermediate step in the mesh generation process, referred to here as CAD-reconstruction, that is illustrated in Figure 3.4. This module first associates the vertices of a linear mesh with the closest BRep objects (surface, edge or vertex) and their coordinates in parametric space. CAD-Conformity of the linear mesh to the CAD geometry is then ensured through vertex projection. A series of logical steps is subsequently employed to reconstruct the CAD connectivity of all surface entities of the mesh, i.e. edges and faces, by associating each mesh object with its corresponding BRep object.

With this newly acquired CAD-mesh information, we reuse and improve the internal bottom-up modules for mesh curving. This allows us to generate valid high-order meshes even from invalid, non-watertight CAD BRep of complex geometries, assuming a valid linear volume mesh is provided, for example generated from a separate triangulation. As this can be reliably achieved by commercial linear mesh generators, the requirements and the need for CAD repair are significantly relaxed.

After establishing this link, accurate parametric projections and surface optimisations are performed for mesh curving of the internal bottom-up pipeline. To ensure validity and to increase the mesh quality, we can optionally employ a variational optimiser to untangle any invalid elements or highly deformed elements. The availability of the CAD connectivity also enables a certain degree of movement of the high-order nodes along the CAD BRep object, which was not previously possible.

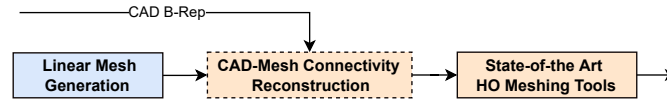


Figure 3.4: The new industrial mesh generation workflow starts from third-party linear meshes (blue) which are processed by in-house *a posteriori* high-order modules (orange).

A special consideration is taken when a linear element (boundary face or edge) crosses two BRep surfaces. In these rare cases, no BRep object is assigned to the high-order mesh entity and is followed by a simplified projection process with no further optimisations. This might result in a slight deterioration of the surface quality in cases where the multi-patch BRep has only C^0 continuity between patches. However, it also allows for significant flexibility on the linear mesh generation side, decreasing further the need for CAD defeaturing and also facilitating the use of two separate geometrical representations for the linear mesh generation (e.g. an STL triangulation) and for the high-order part of the process (invalid STEP).

The CAD-reconstruction is extended to high-order meshes, which have two main advantages. Firstly, it provides the means for a mesh quality suite, where the geometrical accuracy of any high-order mesh can be evaluated, provided a CAD file is present. Secondly, it permits the user to start the workflow from any third-party mesh, even a high-order curved one, assess its quality and then use in-house optimisations to improve it in a geometrically informed manner.

Lastly, we improve the high-order mesh prism layer for maintaining better geometrical accuracy with the desired y^+ resolution as in the original isoparametric boundary layer splitting. However, we also propose a reformulation of this concept to generic high-order mesh modification techniques, both for handling generic unstructured type elements in the viscous layer meshes and for decreasing the manual iterations required for very coarse meshes that are initially unsuitable for mesh curving.

3.3 Workflow Implementation

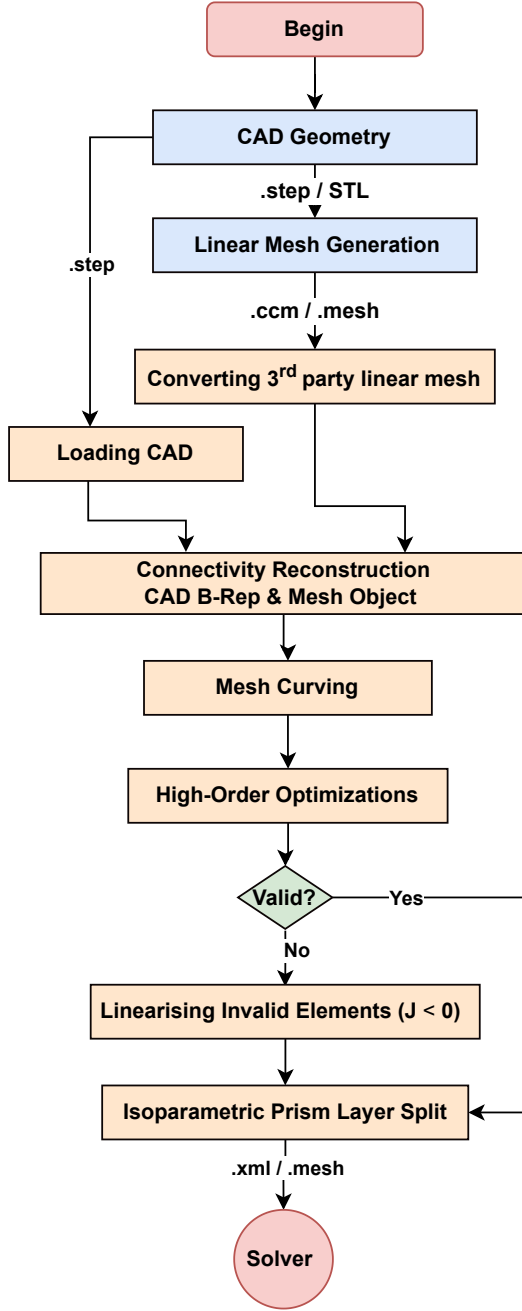


Figure 3.5: Overview of the end-to-end mesh generation pipeline, showing all steps/modules in generating the valid high-order mesh. Processes marked in blue are third-party software, whereas all orange ones are internal to NekMesh.

This work employs, improves and further develops the modules and architecture of the high-order mesh generator NekMesh [35], that has been designed for the generation of unstructured highly-stretched hybrid meshes with primarily prismatic and tetrahedral elements, required for high-speed viscous flow CFD simulations. The section presents the proposed workflow, and discusses all stages of mesh generation, from CAD preparation and linear mesh generation to final untangling. Particular interest will be paid to achieving the required resolution in the normal direction of the prism layers to accurately model the flow in the boundary layer. Figure 3.5 shows all these main processes, part of this industrial pipeline.

3.3.1 Linear Mesh Generation

The starting point of the process is the creation of the underlying CAD model and the generation of the linear straight-sided mesh. The way these are generated is left completely to the desires and needs of the user. However, there are best practices that have been discovered during the development and the applications of this work. In terms of the CAD BRep, there are no specific requirements on the user for the linear generation. The only necessity on the high-order side of the pipeline is the availability of a BRep in the form of a STEP/IGES file. This flexibility allows for linear mesh generation from various CAD sources, including triangulations or implicit representations.

This is advantageous when handling complex geometries because it does allow some imperfections in the BRep. Their presence might lead to a slight deterioration of the surface quality of the mesh, but as described robustness remains priority. As a best practice, it is recommended to use a single thick O-type prismatic or hexahedral that is with sufficient thickness to account for the mesh curving, see Figure 3.6. While not strictly necessary, this streamlines the presented workflow drastically.

Currently, NekMesh supports the input from the following mesh standard formats: CGNS (.cgns), Gmsh (.msh), Nektar++ (.xml) and StarCCM+ through the CCM to OpenFoam API [28] libraries. Supporting other input formats is a relatively straightforward task to implement due to the modularity of NekMesh [70].

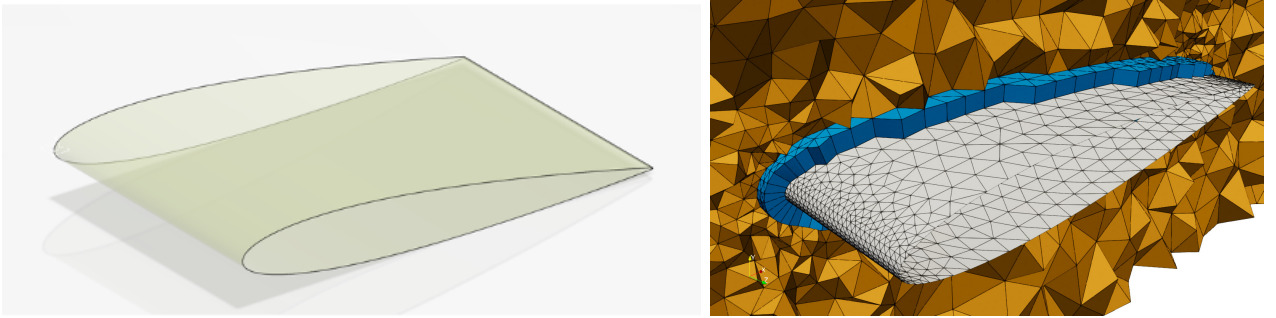


Figure 3.6: The CAD BRep without the domain a). And a cut through view of the straight-sided mesh of the NACA0012 wing b). It displays the surface triangulation (grey), the boundary-layer prisms (blue) and the tetrahedra outside the boundary layer (brown).

3.3.2 CAD-reconstruction

This section describes a novel method for reconstructing the lost CAD mesh connectivity which combines bottom-up approaches for vertex BRep association through the CAD API with a set of top-down logical operations derived from the BRep topology. This module is designed to be applied on complex geometries and to be compatible with any linear mesh generator. To achieve this we have eliminated a large number of restrictions and assumptions of the legacy processes. For example, it allows for lack of CAD conformity of the vertices and for the presence of multi-patch elements. Moreover, the user can provide usually invalid CAD. Further, the new modules of NekMesh permit a full integration with third-party curved meshes. The final outcome of the process is the connectivity between the BRep and the mesh objects, as illustrated in Table 3.1.

Table 3.1: The final association between BRep and mesh objects.

Mesh/CAD	Vertex	Edge	Face	HO-EdgeNode	HO-FaceNode
CAD-Vertex	✓	✗	✗	✗	✗
CAD-Curve	✓	✓	✗	✓	✗
CAD-Surface	✓	✓	✓	✓	✓

Last, a high-order summary of the module can be seen in Algorithm 1.

Algorithm 1 CAD-Reconstruction

In: Mesh $\mathcal{M}$, CAD-BRep (STP/IGES)

Out: Bnd Conforming $\mathcal{M}$ + CAD-mesh link

```

1: Load CAD
2: Create Bounding Boxes - CADSurf, CADCurve, and CADVert
3: Link bnd vertices to CAD objects, ( $\epsilon_{loose}$ )
4: Project bnd vertices to closest CAD object
5: Link bnd vertices to CAD objects, ( $\epsilon_{tight}$ )
6: Link mesh faces to a CADSurf
7: Link mesh edges to a CADCurv/Surf
8: if High-Order (HO) nodes exist then
9:   Link HO-nodes to parent edge CAD
10:  Link HO-nodes to parent face CAD
11: end if
12: if edge !CADObject then
13:   Project multi-patch edges to closest CADSurf (node-by-node)
14: end if

```

BRep and CAD API

The CAD APIs handle the geometric information provided in Boundary Representation (BRep) format, typically via STEP or IGES files. As the linear volume mesh is already available before the curving stage, the existing mesh connectivity data can be leveraged to relax traditionally strict geometric requirements imposed on STEP. This flexibility significantly improves robustness against common STEP-related artifacts, such as free edges, pierced surfaces, non-manifold edges and other topological errors that typically pose challenges in the initial volume meshing. Moreover, a user can provide a non-watertight geometry covering only certain parts of interest of the body for mesh curving. One can also reuse the geometry multiple times by excluding the outer CFD domain from the BRep, as per the NACA0012 in Figure 3.6. The only requirement for the BRep surfaces is a valid underlying parametric definitions.

All geometrical operations and queries performed within NekMesh rely on dedicated CAD APIs, which interface directly with established CAD kernels. Currently, NekMesh supports two distinct CAD engines: OpenCascade [13] and ITI CADfix [39]. In this study, the OpenCascade one was used. While some extensions were part of this work, we refer to the one by Turner [106] and Marcon [62] for the details of these CAD APIs.

Bottom-up BRep Pre-Processing

The vertices of the initial linear mesh are the only mesh entity sufficiently close to the BRep. Hence, the first step of the reconstruction process is to associate every boundary vertex of the linear mesh with the CAD surfaces, CAD Curves and CAD vertices. A naive approach that would query the distance to every BRep object is unfeasible for industrial geometries that may contain more than 10^4 surfaces. To reduce the cost of such queries, a preprocessing of the BRep is first performed. All CAD surfaces are sampled with nodes from which their bounding boxes are calculated and are enlarged. Then all these bounding boxes are stored in a k-d tree, as in the original NekMesh workflow [104]. Here we extend this procedure to CAD curves and every CAD vertices, all stored in three separate k-d trees.

Vertex Associations

This way every boundary vertex can be efficiently associated with only a few candidate surfaces when it is inside their bounding box, see Figure 3.7. These can vary from 3 to 20 per vertex roughly depending on the CAD complexity. The vertex is then virtually projected only onto the candidate surfaces, the surfaces are ordered according to the distance δ , and discard those where $\delta > \delta_{tol}$. A default value used for that is $\delta_{tol} = \max(0.5L, 10^{-5})$ where L is the length of the edge, but these can be adjusted depending on the application. If more than one CAD Surfaces is shortlisted, we proceed with a look-up for CAD Curves and CAD Vertex. Finally, we project the vertex to the closest CAD Curve or CAD surface. To ensure even further reduction of false positive associations, we run the same cycle of associations without the final projection with a 10 times stricter tolerance and reevaluate the list with the parametric location for all sufficiently close candidate CAD objects. This way we ensure the CAD conformity of the linear mesh and a minimal amount of false positives, see Figure 3.8

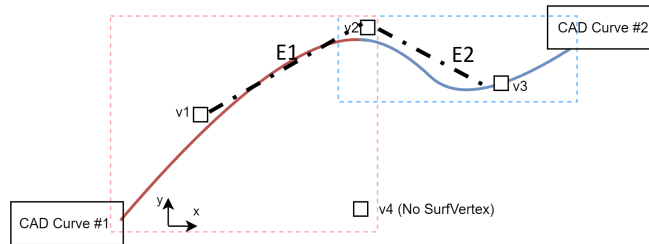


Figure 3.7: Diagram showing the process of vertex association to BRep objects before projection. Note that none of the vertices is exactly on the BRep. Vertices V_1 and V_3 will have only one associated CAD curve, whereas V_2 will have two.

Face Associations

Now that the mesh vertices have been linked to their corresponding CAD objects, we move forward to boundary faces. From a CAD connectivity standpoint, boundary faces are classified into three distinct types: internal elements, corner elements (also referred to as Case 2), and multi-patch elements spanning over two CAD surfaces (Case 3). These cases are illustrated in Figure 3.8 with a triangular face, but the implementation is applied for both triangular and quadrilateral faces. Thus achieving a full coverage for all common mesh elements types.

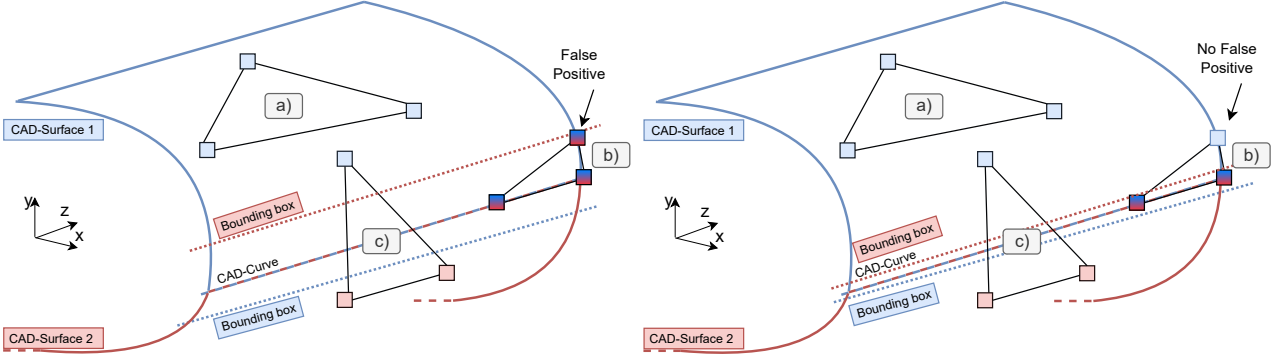


Figure 3.8: Diagram showing the 3 different element cases: a) internal face, b) corner (Case 2), and c) multi-patch (Case 3) with initial relaxed and consequently strict association tolerances.

The CAD-reconstruction for “internal” elements is the easiest as all its vertices share a single common surface. In addition, at least one vertex is associated only with the common surface. These two conditions are usually met by more than 95% of the boundary elements, enabling for better parametric mesh curving, surface and volume optimisations.

Case 2 elements, with vertices having more than one common CAD surface, are more difficult to handle. These often appear when one of the surfaces is very small or with a very high aspect ratio like, for example in blunt trailing edges or sliver surfaces as illustrated in Figure 3.9. In this case, the algorithm samples the mid-point on all edges and the centroid of the face, calculates the distance to the candidate CAD surfaces with the CAD API and then picks the one with the minimum overall deviation. The surface ID is obtained as follows

$$\text{ID} = \arg \min_k \sum_{j=1}^3 \|\mathbf{x}_j - \mathbf{P}_k(\mathbf{x}_j)\| \quad (3.1)$$

where $\mathbf{P}_k(\mathbf{x}_j)$ denotes the coordinates of the projection of the point $\mathbf{x}_j$ on the surface of ID k . This is effective for both straight-sided and curved surfaces, given a with a reasonably deviation from their linear approximation, which we allow to be $\delta_{\max} = 0.5 \times L_{\text{edge}}$.

Finally, if boundary face vertices do not share any common CAD surfaces or the element fails any of the previously described filters, no BRep object is associated with the element, Case 3

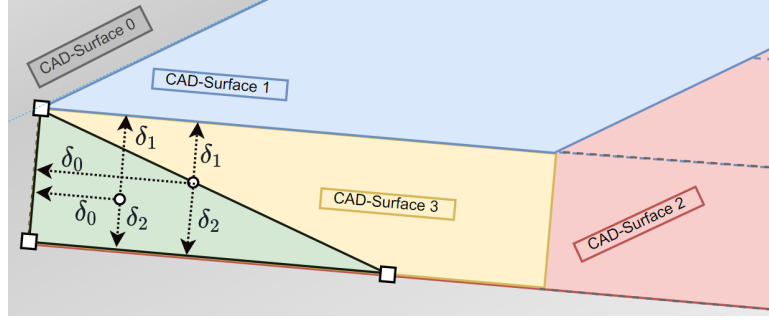


Figure 3.9: Specific treatment of the objects with multiple CAD surface candidates in a region of trailing edge or corner elements (Case 2).

face. Hence, a simpler mesh curving algorithm is applied to this element without local surface optimisation and these faces do not slide on the CAD during the global volume optimisation.

Edge Associations

The edge association to CAD surfaces is relatively straightforward. By revisiting Figure 3.7, it can be observed that when the vertices have a single associated common CAD Curve or CAD Surface, this edge is internal to the BRep object and can be directly linked to it. In the case of CAD Surface association, a cross-check is performed with the parent element CAD surface. A priority is given to the CAD Curve link.

If the vertices share two or more CAD surfaces, but not a CAD Curve, we apply a redundancy association approach. Topologically, this indicates a corner edge likely generated on a CAD curve. Therefore, we loop over all curves, forming the EdgeLoops around the candidate CAD surfaces creating a shortlist of CAD curve candidates. If the edge has a single common one, it is connected to this CAD curve. If two curves are present (around junctions), another voting process is performed, choosing the closest one within a tolerance of δ_{tol} . In the case of more than two common CAD curves, we prefer the safer choice of assigning the elemental CAD surface to the edge to facilitate a more accurate projection and surface optimisation. Lastly, if the vertices of the edge do not share a CAD surface or fail the CAD curve tolerance test, this indicates that the edge is crossing multiple patches. Hence no BRep object is associated at this stage, and a simpler curving process is applied as per the Case 3 elements.

Extension to High-Order Meshes

Once the mesh-CAD connectivity is available for the linear mesh, we can extend the method to high-order mesh entities such as edge nodes and face nodes. This is straightforward since the edge nodes will adopt the CAD curve or CAD surface of the parent edge, and the face nodes the CAD surface of the parent face. Interestingly, this brings about the unique workflow

capability to NekMesh side from any point of the process. Moreover, it allows to restart from any kind of a third-party mesh, being it linear or high-order. To the best of our knowledge, this is a unique capability and has not been reported previously.

3.3.3 Mesh Curving and Surface Quality

Once the connectivity between the linear mesh and the CAD BRep is established, we can apply any high-order mesh curving module that uses the CAD BRep. In this workflow, we modify and enhance the interpolative mesh curving algorithm, which is based on the parametric projection of high-order nodes onto the BRep, followed by surface mesh optimization, part of NekMesh [35]. This has already been proven to produce high-quality meshes in the in-house bottom-up mesh generation workflow [106, 16] that generates hybrid tetrahedral and prism-tet linear meshes. Moreover, we extend that capability for also fully hexahedral or hex-dominant meshes.

The mesh curving starts with the parametric projection, which is applied hierarchically for every boundary face in the parametric space of the corresponding CAD object (u, v) . First the edges are curved, followed by the internal face nodes. When we consider the edges, the high-order nodes are created in the reference space according to the user-defined order and quadrature rule, usually Gauss-Lobatto-Legendre. Then following the edge vertices in (u, v) or (t) the nodes are mapped to the parametric space. Given the (u, v) coordinates of every high-order edge node, we use the CAD API to map back its location back to Cartesian coordinates (x, y, z) . The process is illustrated in Figure 3.10.

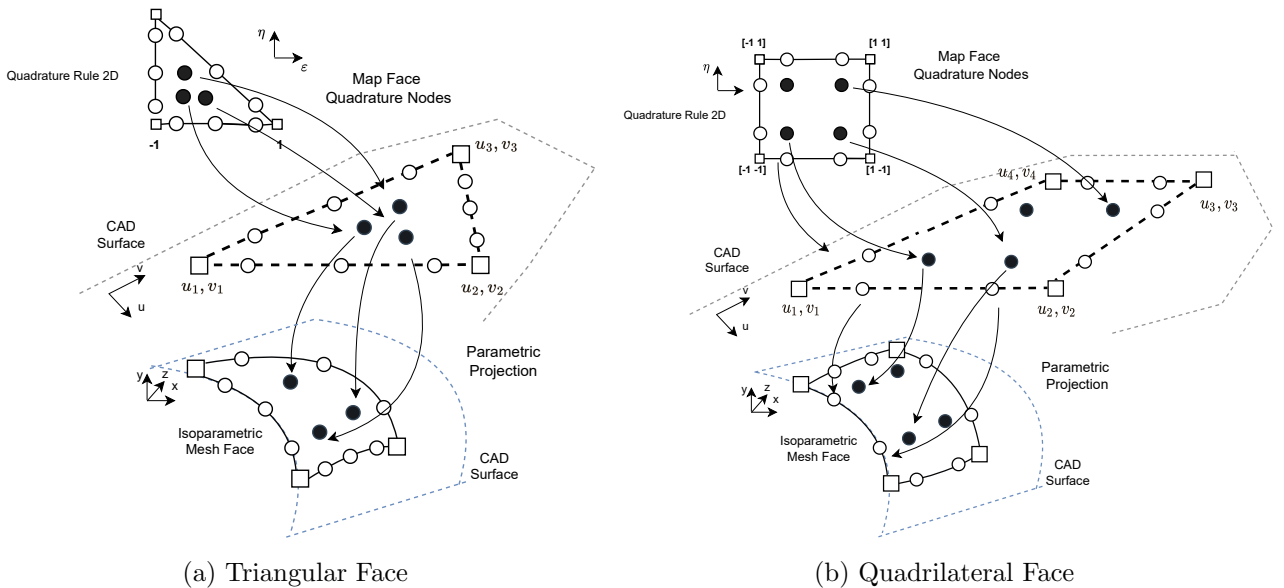


Figure 3.10: The 3D parametric projection of the high-order face nodes in parametric (u, v) to physical space for triangular $p=4$ and quadrilateral $p3$ faces).

After the parametric projection of the edge nodes, we apply the same type of parametric projection on the corresponding internal face nodes, see Figure 3.10. Similarly, the desired quadrature nodes are generated according to the user-defined order and quadrature, then mapped to the parametric space of the corresponding BRep surface with the parametric knowledge of the vertices. Last, these are projected on the CAD and the validity of the element is checked. This way, the workflow provides very high geometrical surface accuracy with the user-defined quadrature order, tested up to polynomial order $p = 14$.

In the rare cases where the boundary face does not possess a CAD surface association, i.e. they are not CAD compliant, we rely on a simplified secondary mesh curving process. In this case, the face has at least one edge crossing two BRep surfaces. As discussed earlier, this is not a desirable situation where the exact parametric projection is not possible. However, when faced with BRep slivers or meshing from triangulations, it is crucial to account for them and devise strategies to avoid having linear or tangled elements. Therefore, we populate the desired quadrature on the linear element and then project the high-order edge nodes one by one to the closest CAD surface, as depicted in Figure 3.11. This fail-safe process relies on the curved edges to reconstruct the underlying geometry for robustness purposes. The reason being that projecting high-order face nodes on the closest CAD surface is more likely to result in self-intersections, especially when the two surfaces have high curvature and are only C^0 continuous. No surface optimisation is applied to these mesh faces.

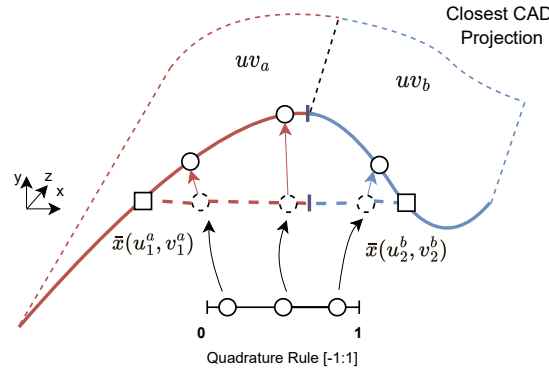


Figure 3.11: Diagram showing the 3D projection of edge nodes from Cartesian space (x, y, z) onto the closest CAD surface.

3.4 Surface Mesh Optimizations

During the mesh curving, the workflow moves the high-order nodes on the BRep object to minimise the distortion induced by the parametric projection according to the optimization method described in references [97, 105]. On an edge, we achieve that by introducing a virtual spring network system between the edge nodes and moving them on the BRep object to minimise the

potential energy

$$F = \sum_{i=1}^P \frac{\|\mathbf{x}(u_{i+1}, v_{i+1}) - \mathbf{x}(u_i, v_i)\|^2}{w_i}. \quad (3.2)$$

where the w_i weights are purely geometrical - with $w_i = \xi_{i+1} - \xi_i$, where ξ_i is the node location in the reference space as defined by the nodal distribution. A minimum of this energy is calculated in Cartesian space using a classical BFGS optimization algorithm [12]. This process becomes important when coarse elements are present in regions with very high two-directional curvature, such as wingtips or rounded trailing edges. In these cases, we observe a significant increase in the surface mesh quality, geometrical accuracy and even untangling of self-intersections, as shown in Figure 3.12. Naturally, this translates to an improved overall volume mesh quality.

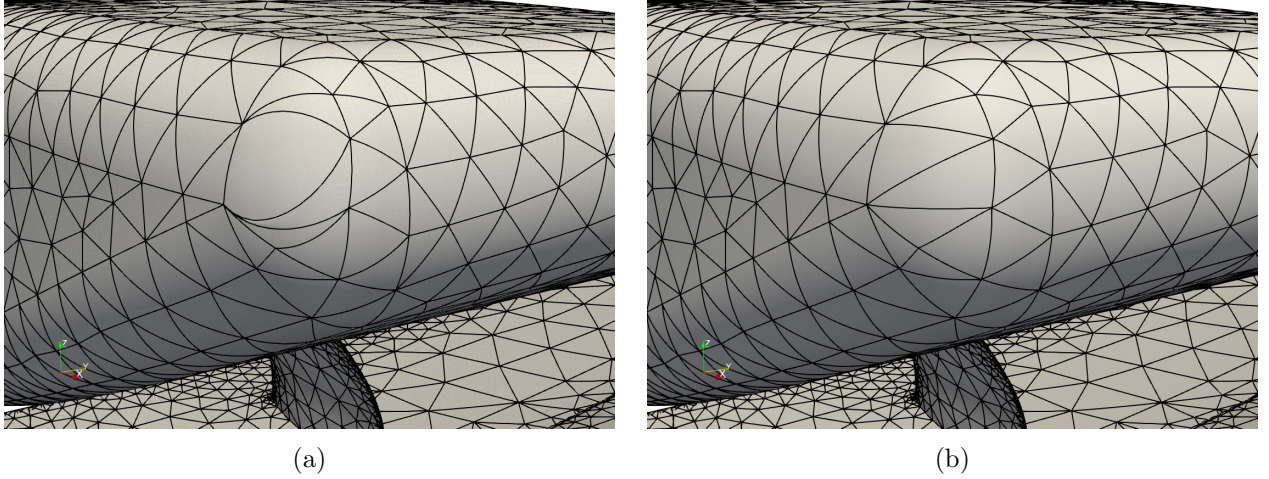


Figure 3.12: The curved surface mesh generated with (b) and without (a) the optimisation. Note the presence of invalid elements when the optimisation is not applied.

3.4.1 Assessing Geometrical Accuracy through CAD-Reconstruction

The geometrical errors in the mesh discretization of the underlying CAD could have a significant impact on the accuracy of simulations, especially for boundary layers at high flow speeds. Often it is difficult to identify these errors. The high-order extension of the CAD-Reconstruction algorithm provides a tool for assessing the geometrical accuracy of the mesh, which can be quantified by the distance between the node and the target CAD-object $\delta \mathbf{x}_i$, and by an angular error between the node normal and the corresponding CAD-Normal $\delta \mathbf{n}_i$.

The first stage of the process is to elevate (sample) the boundary face to an order of choice with the desired quadrature points on the face. Then the CAD connectivity is reconstructed and all vertices, high-order edges and face nodes receive their BRep parametric coordinates (u_i, v_i) . This backward mapping allows us to then query through the CAD-API for the target

location $\mathbf{x}_i$ and the unit normal vector $\mathbf{n}_i$ on the associated CAD object. Last, we calculate the corresponding nodal deviations for $\mathbf{x}_i$ and $\mathbf{n}_i$ between the isoparametric and BRep nodes, see Figure 3.13. Through the Nektar++ high-order utilities, we can visualise the elemental values of these deviations and assess if any further interventions are necessary. This also allows the user to visualise and investigate if, where and why an element does not have CAD associated with it which will indicate the presence of a multi-patch face or a rare failure of the CAD-Reconstruction algorithm.

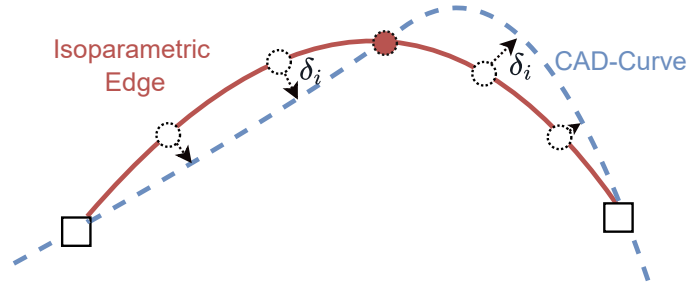


Figure 3.13: Diagram showing the mechanism of assessing the geometrical accuracy $\delta\mathbf{x}_i$ of a $p = 2$ mesh evaluated at $p = 6$ with an evenly distributed quadrature rule.

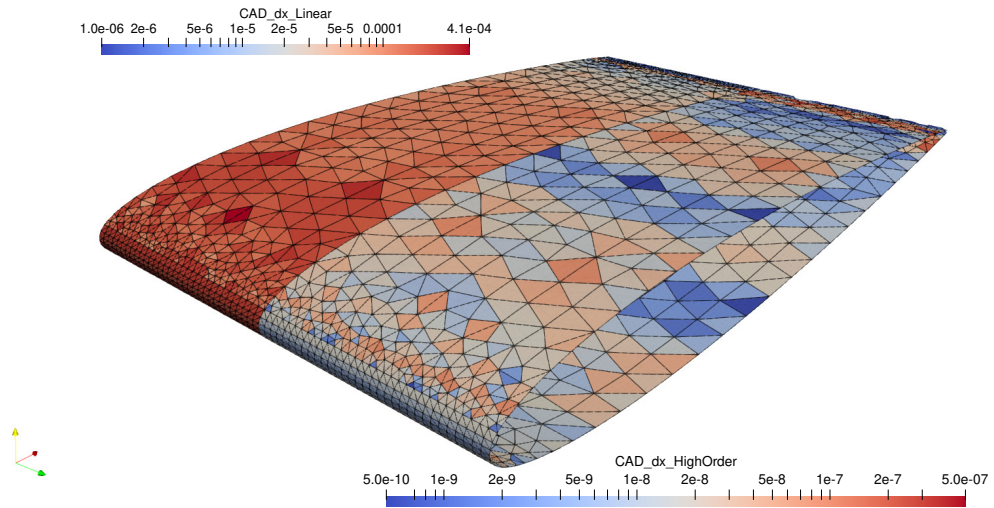


Figure 3.14: Measures of geometrical accuracy ($\delta_{\max}$) for the NACA0012 displayed on the linear (left) and the curvilinear (right) surface meshes. Note the different scales of the top (linear) and bottom (curvilinear) legends.

3.4.2 Variational Optimizer (Volume Optimization)

To improve the mesh quality and untangle invalid elements, we (re)use the variational optimiser module in the NekMesh framework [105]. This is achieved through the minimisation of the deformation energy functional W , which is a function of the mapping ϕ from the ideal straight-sided element Ω_I^e to the curvilinear element Ω^e , formulated as

$$\min_{\phi} \mathcal{E}_i(\nabla\phi) = \sum_{e \in i} \int_{\Omega_I^e} W(\nabla\phi) \, dy. \quad (3.3)$$

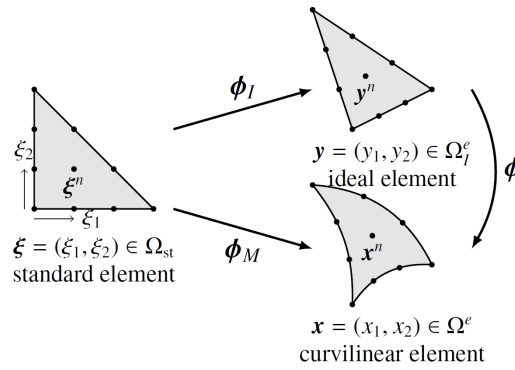


Figure 3.15: The mappings ϕ from standard reference element Ω_{st} to ideal straight-sided Ω_I^e and the final curved element Ω^e . Note the mapping used for optimization is $\phi = \phi_M \circ \phi_I^{-1}$ [73].

The various mappings involved in this setting are depicted in Figure 3.15. The optimization calculates the positions of the nodes that lead to a minimum of the energy, see Equation 3.3. We direct you to reference [106] for a detailed description of the method.

The energy functional adopted here is based on hyperelastic materials

$$W = \frac{\mu}{2}(\text{tr}(\mathbf{C}) - 3) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2 \quad (3.4)$$

where $\nabla\phi(\mathbf{y})$ is the gradient tensor of the mapping, its determinant $J = \det(\nabla\phi)$ is denoted the Jacobian, λ and μ are the Lamé elastic constants, and $\mathbf{C}$ is the Cauchy-Green stress tensor.

The untangling capability of the method is achieved through the regularisation of the Jacobian for values $J < 0$, and is significantly improved by allowing the movement of the nodes on the BRep object except for the mesh vertices associated with the CAD vertices. A minor limitation of our process is that we guarantee perfect CAD surface, but not perfect CAD curve and CAD vertex association. Therefore, for robustness, we also fix the nodes on the boundary associated with two or more surfaces, two or more curves and the ones without a CAD object.

3.5 An Improved Isoparametric Boundary-layer Splitting

Very high-aspect ratio elements are required near the boundary for viscous flow simulations. The main challenge is to propagate the curvature inside the domain and ensure the quality and validity of the final mesh. For this purpose, we employ the standard NekMesh approach for generating and curving a thick single prism layer around the body, the macro-prism layer [73]. Every macro-prism is mapped to standard space and it is split according to a user-defined progression ratio (r) and a number of boundary layers. This work introduces an improved implementation in NekMesh by propagating not only the edge curvature through the edge nodes but also all face nodes, hence providing higher surface accuracy.



(a) Valid curved high-order macro-prism before splitting. (b) Isoparametric splitting in reference space with $N = 6$ prism layers and a progression ratio $r = 1.5$.

Figure 3.16: Isoparametric boundary layer splitting in the reference space (ξ) and the mappings from reference to Cartesian space [73].

Chapter 4

Applications to Industrial Geometries

To demonstrate the quality and effectiveness of the proposed meshing technique, we present three examples of increasing geometrical complexity: one academic and two automotive. These include a simple domain around a hemisphere, an inverted multi-element extruded wing to show the flexibility of the approach and the full version of the Imperial Front Wing as a complex motorsport geometry [82, 11] representative of an industrial application.

4.1 Consistency - A Semi-spherical Geometry Test Case

This elementary geometry is a domain around a hemisphere with a BRep created by four NURBS instead of an analytical surface in an industrial CAD software. The process works as follows. A very coarse linear mesh is generated from STEP using a third-party mesher, with only 4 prism elements representing the hemisphere, see Figure 4.1a. The CAD-Reconstruction process successfully associated the necessary connectivity between all the mesh elements and their associated BRep objects, and performed parametric projections with surface optimisation, as described in Section 3, to polynomial order $p = 9$ producing the mesh shown in Figure 4.1b.

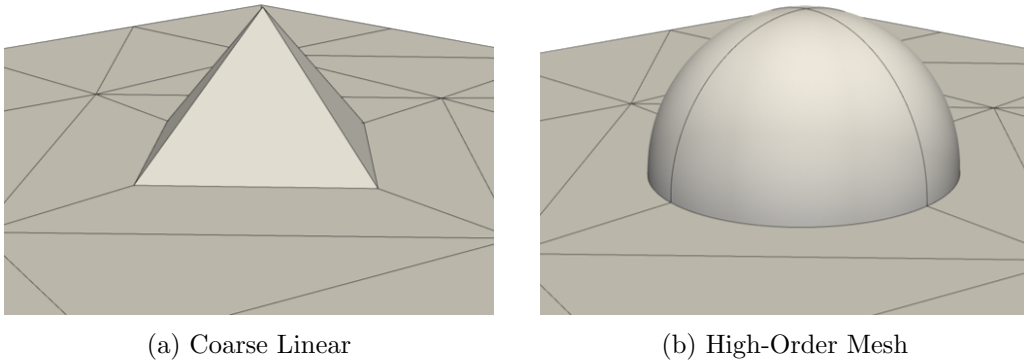


Figure 4.1: The high-order mesh generation of a domain around a hemisphere: (a) Initial linear mesh , and (b) the curved surface mesh optimised at polynomial order $p = 9$.

Given the analytical nature of the geometry, we can evaluate the surface mesh geometrical accuracy against the CAD model but also evaluate the error of the model itself against the analytical hemisphere, see Figure 4.2 a). The CAD BRep we used had a maximum error of $\delta_{CAD} = 3.2 \times 10^{-5}$ located around the base and the apex, likely attributed to the use of non-analytic CAD surfaces generated with interpolative or approximation operators (e.g., Coons patches), commonly used within CAE workflows. Our mesh shows a similar error distribution to the CAD reference with slightly smaller δ_{max} against the analytical hemisphere, but also marginally increased δ around edge, see Figure 4.2 b).

To quantify geometric accuracy against the CAD model, we use the proposed quality suite. Each $p=9$ isoparametric element is sampled with evenly spaced interpolation points ($N=16$ and $N=21$). This high sample number was chosen due to the non-uniform distribution of the δ inside the element and the sensitivity to their location. Then the deviations δ are computed against the parent CAD object, see Figure 4.2c). The analysis revealed negligible deviations for the element edge nodes, while maximum ones concentrated primarily around face nodes close to the element edges, likely due to the $p=9$ and the subsequent Runge Phenomenon. Overall, the maximum deviation is $\delta_{max} = 9 \times 10^{-6}$, roughly three times lower than the approximation error of the original B-Rep model. Furthermore, 74% of the sampled high-order nodes exhibited deviations of less than $\delta = 3 \times 10^{-6}$, demonstrating the workflow's capability to accurately represent the CAD BRep to values comparable to or even lower than the original CAD model with very coarse meshes utilizing high polynomial orders.

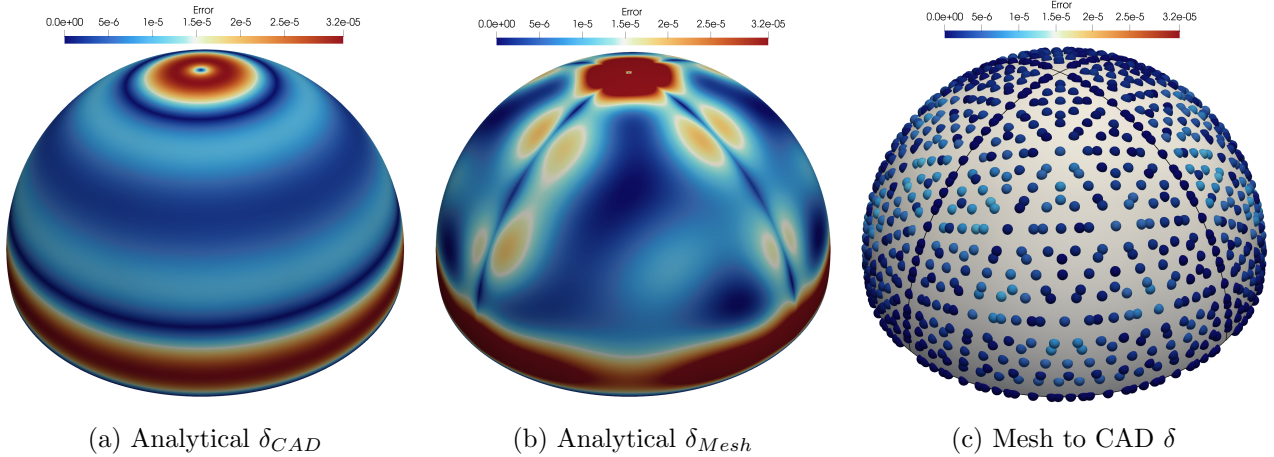


Figure 4.2: Geometrical error of the CAD model (a) and the $p=9$ HO-mesh (b) against the analytical hemisphere showing comparable max errors at the base and the apex with slightly increased δ around the mesh edges. And the geometrical accuracy of the same $p=9$ mesh against the CAD (c), evaluated at evenly spaced points on the HO-mesh, displaying δ_{max} against the CAD being three times lower than the CAD model approximation error.

Using the same geometry but with a finer all-tetrahedral linear mesh created separately in NekMesh, we verify the consistency between the existing mesh generation workflow, as implemented by Turner [106], and the new proposed workflow. The linear mesh is depicted in

Figure 4.3a. The process successfully reconstructed all the CAD objects: 1650 vertices, 2474 edges, 1650 faces, and 6908 elements. This allowed accurate parametric mesh curving, surface optimisation, and the variational optimiser with sliding of the internal to the CAD surfaces mesh nodes and vertices. As the histogram of Figure 4.4a shows, the initial curved mesh is valid but certain elements have low quality in terms of scaled Jacobian, with the minimum value $J_{\min}=0.36$. After the application of the variational optimiser, see Figure 4.4b, the overall quality of the mesh increases drastically, with a minimum scaled Jacobian $J_{\min} = 0.75$. Both the geometrical accuracy and the final mesh quality are similar to the legacy internal bottom-up mesh generation workflow with $J_{\min} = 0.74$, which is a difference of 1%. This shows the consistency of the proposed methodology with the high-quality internal bottom-up one, which is to be expected due to the reuse of the high-order modules and the successful CAD-Reconstruction.

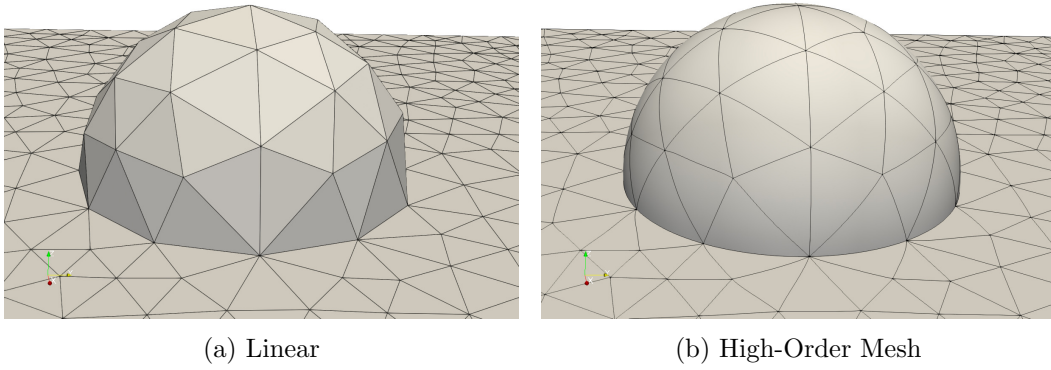


Figure 4.3: The high-order mesh generation of a domain around a semi-sphere starting from a linear mesh (a) and the curved surface mesh (b).

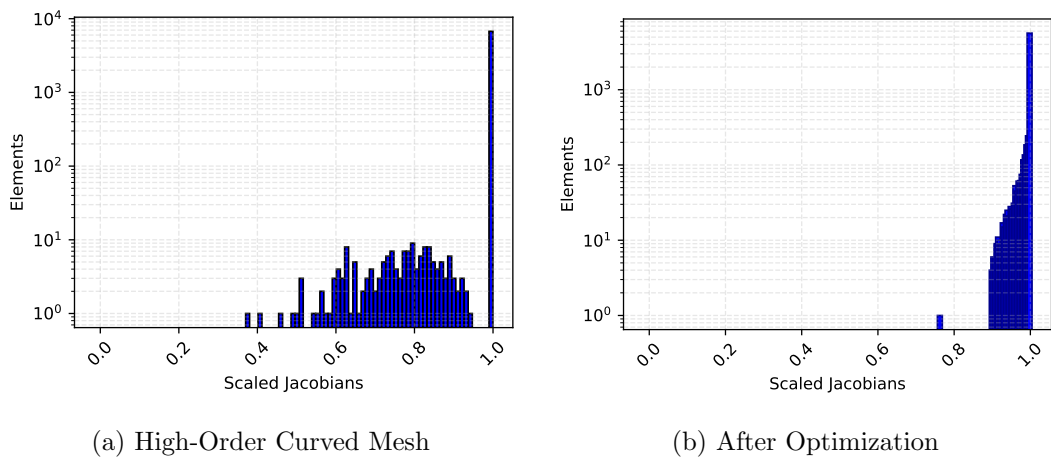


Figure 4.4: The mesh quality of the high-order mesh generation of a domain around a semi-sphere: (a) before and (b) after variational optimisation.

4.2 Imperial Front Wing

This workflow is aimed at the robust and automatic generation of high-order meshes for high-fidelity simulations of flow about complex geometries. We demonstrate the suitability of the workflow from a design perspective using an industrial example of such geometry: the front wing of a motorsport car. The Imperial Front Wing, is an open geometry from a legacy F1 car designed by McLaren Racing [11]. The front wing, shown in Figure 4.5, consists of a main plane, two wing flaps, a nose cone, a Gurney flap, a canard, a side endplate with a connected footplate.

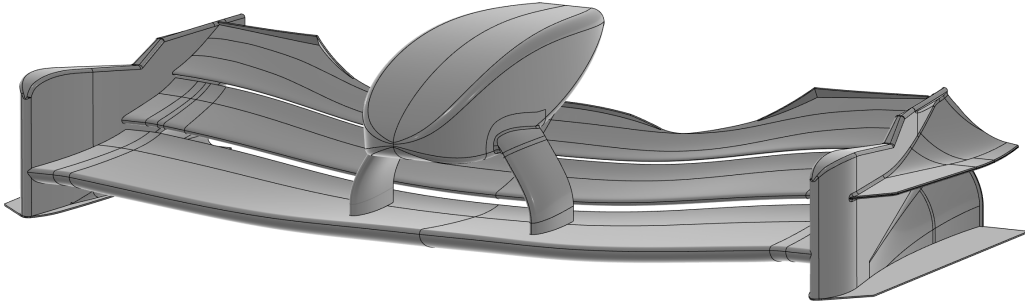


Figure 4.5: The CAD BRep geometry of the Imperial Front Wing (IFW).

The principal characteristics of the STEP BRep geometry are given in Table 4.1. Based on this information, the geometry can be thought of having medium complexity, but it is sufficiently complex to pinpoint many of the previously described problems arising from industrial CAD such as CAD slivers, parametric mesh curving distortions and the presence of BRep curves in close proximity requiring the use of multi-patch elements, see Figures 2.8 and 3.12.

Table 4.1: Statistics on the BRep of the Imperial Front Wing.

CAD Object	Count	Parameter	Value [m]
Vertices	351	$L_{\min \text{ edge}}$	0.4×10^{-3}
Curves	559	$L_{\text{main plane}}$	0.2
Surfaces	209	$R_{\min \text{ curvature}}$	0.2×10^{-3}

To allow for a quicker testing of numerical models and benchmark performance, a simplified 3D geometry of the IFW was created by extruding the cross-sectional slice at the mid-span location of the IFW. The 3D extruded Imperial Front Wing with $L = 0.05$ thickness, referred to as “eIFW” in the following, will also be used in this study for verification and testing purposes.

Mesh Flexibility - 3D Extruded Imperial Front Wing

This section uses the simplified extruded wing eIFW to demonstrate the flexibility of the proposed workflow in handling different types of elements: tetrahedral, hexahedral, and hybrid prismatic-tetrahedral meshes. Moreover, the straight-sided volume meshes are generated in three different software: NekMesh for the linear tetrahedral meshes, a combination of Gmsh and NekMesh for the fully hexahedral one and StarCCM+ for the hybrid mesh. The linear vertices of the NekMesh tetrahedral mesh is on the CAD, whereas the StarCCM+ and Gmsh ones have a maximum deviation of $\delta_{\max} = 2 \times 10^{-6}$ and $\delta_{\max} = 4 \times 10^{-7}$ respectively. The CAD-mesh connectivity is neither retained nor provided *a priori*. Despite the lack of CAD-mesh connectivity, the CAD-reconstruction is successfully applied to all three meshes. Mesh curving and surface optimization are subsequently performed to polynomial order $p = 4$ with Gauss-Lobatto-Legendre quadrature. A comparison between the initial straight-sided and resulting curvilinear surface meshes is shown in Figure 4.7. The mesh quality of the final high-order meshes can be seen in Figure 4.6 with a summary given in Table 4.2. From these, we can observe a good final mesh quality for all meshes $\min J_{sc} > 0.4$, with the hybrid prism-tetrahedral one showing the highest $\min J_{sc} = 0.68$ and not a single highly deformed element ($J_{sc} < 0.5$). This is despite the highly non-uniform curvature of the two flap leading edges being resolved by only two to four elements.

Table 4.2: Comparison of high-order meshes of different types for the eIFW

Mesh Type	Elements	$\min J_{sc}$	$J_{sc} < 0.5$	Comment
Tetrahedral	50750	0.42	1	
Hexahedral	25140	0.44	20	No Volume Optimisation
Prism-Tet	21885	0.68	0	
HO Tet	88054	0.32	4	58 $J_{sc} < 0$ on input (GMSH)

Last, we demonstrate the ability to import and optimize high-order meshes directly from external mesh generators. For this purpose, we use a third-order high-order curved mesh created with Gmsh without optimizing it there. Initially, the mesh contains 58 elements where $J_{sc} < 0$, making the mesh invalid. We then apply the CAD-reconstruction recreating the parametrization also of the high-order nodes. This allows our variational optimizer to effectively untangle the mesh helped by sliding the surface nodes on CAD surfaces. The final mesh is valid with $\min J_{sc} = 0.32$ and only 4 highly deformed elements. Additionally, we observe an improvement of the geometrical accuracy on the second flap of the wing.

Finally, the proposed workflow has been used to generate multiple hybrid prism-tet high-order meshes of the extruded IFW that have been successfully utilised to perform incompressible large-eddy simulations by Liosi, Khurana and Wüstenberg [53, 45, 111] with Nektar++. Their simulations are in good agreement with the large eddy simulations by Slaughter [102] and also with simulations by other flow solvers such as CharLES and Horses3D [79, 76]. Several positive

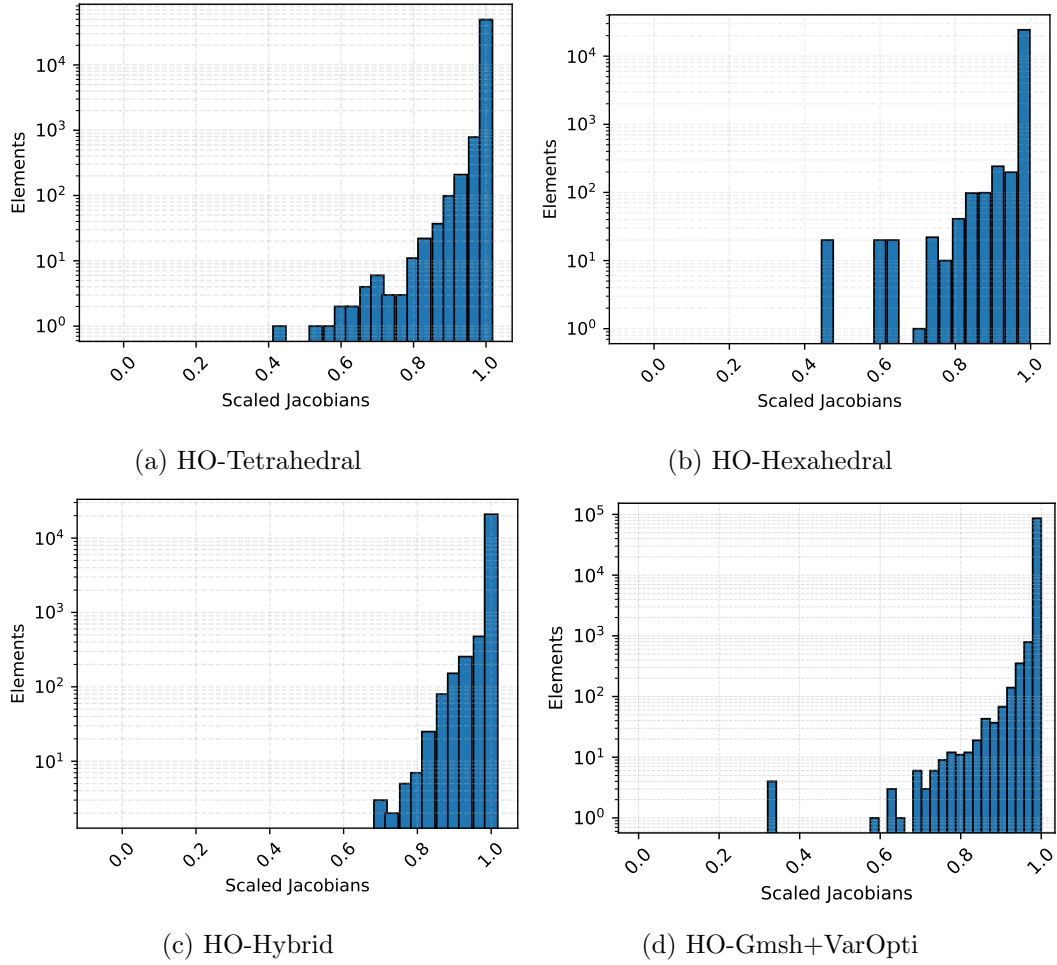
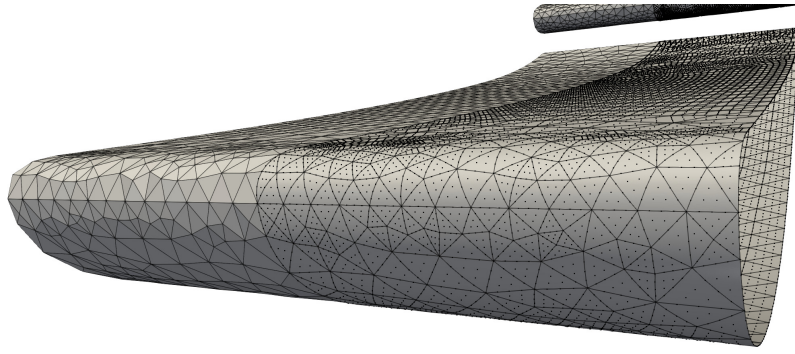


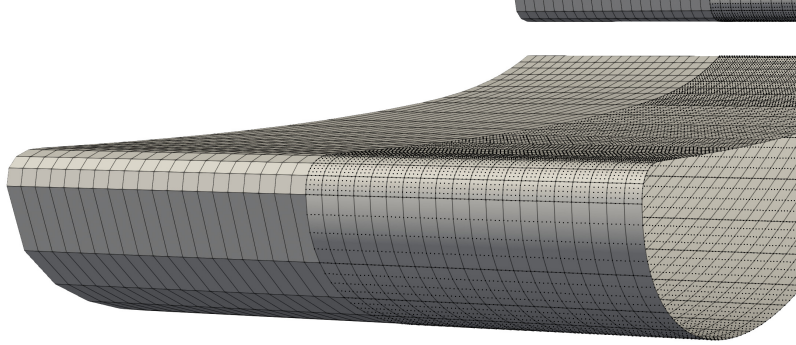
Figure 4.6: High-order mesh quality distribution for the final optimized four variants of the eIFW, evaluated through the scaled Jacobian.

effects were observed with our new meshes during the simulation campaign. First and foremost, the improved geometrical accuracy enabled the correct capture of the transition mechanism through a short laminar separation bubble. Smooth shear stresses were observed without the previously present non-physical fishnet-like pattern of increased τ_{xz} that occurred due to C^1 discontinuities between the curved faces of legacy meshes. This was of great importance for the project's success, as the legacy fishnet effect resulted in an unphysical transition. Second, the improved robustness of the pipeline did not generate any invalid elements for the eIFW, even without the use of the variational optimiser.

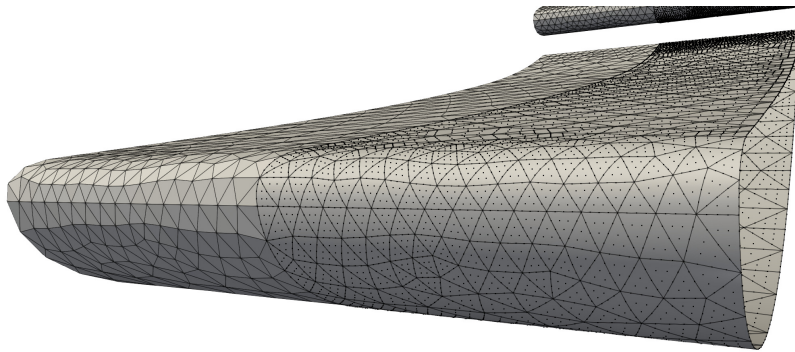
In conclusion, the eIFW case demonstrated that our method handles all common unstructured meshes, manages to reconstruct the CAD-mesh link, curves the linear meshes and produces a final mesh suitable for simulations from both commercial and open-source third-party sources. Moreover, the process can be restarted, can improve and even untangle third-party high-order curved meshes.



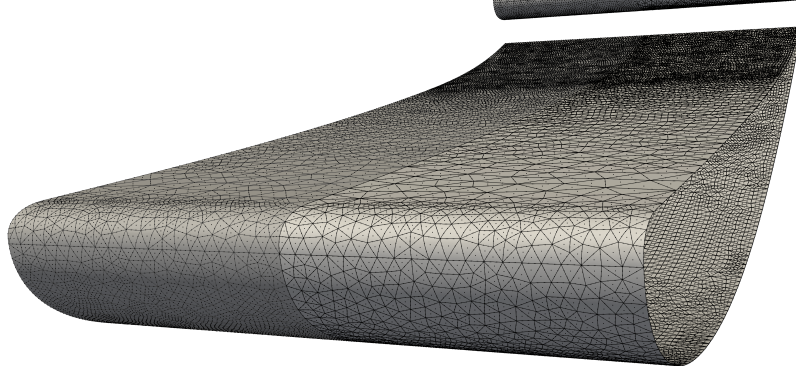
(a) Tetrahedral mesh. Linear mesh generator: NekMesh.



(b) Hexahedral mesh. Linear mesh generator: 2D Gmsh + 3D extrusion: NekMesh



(c) Prism-tetrahedral mesh. Linear mesh generator: StarCCM+.



(d) HO-Tetrahedral. HO-mesh generator: Gmsh + NekMesh VarOpti

Figure 4.7: The 3D linear/initial (left) and high-order $p = 4$ surface mesh (right) of a fully tetrahedral, hexahedral and hybrid prism-tetrahedral meshes, respectively. Last a $p = 3$ invalid ($J_{sc} < 0$) high-order Gmsh mesh improved with the proposed NekMesh pipeline.

Industrial Validation - Imperial Front Wing

The first industrial test case is the full Imperial Front Wing. First, a linear straight-sided mesh is generated directly from the STEP BRep in the third-party finite volume mesh generator, StarCCM+, see Figure 4.8. No CAD defeaturing or simplifications are applied, and CAD compliance is imposed on the whole geometry to demonstrate the reconstruction algorithm. The only region where this is not enforced is the leading edge of the endplate to avoid sliver elements, see Figure 2.8. A single thick prismatic layer is generated near the body, which is sufficiently thick to accommodate the mesh curving process, and tetrahedral elements are used in the domain outside the boundary layer. Additional information about the mesh entities can be found in Table 4.3.

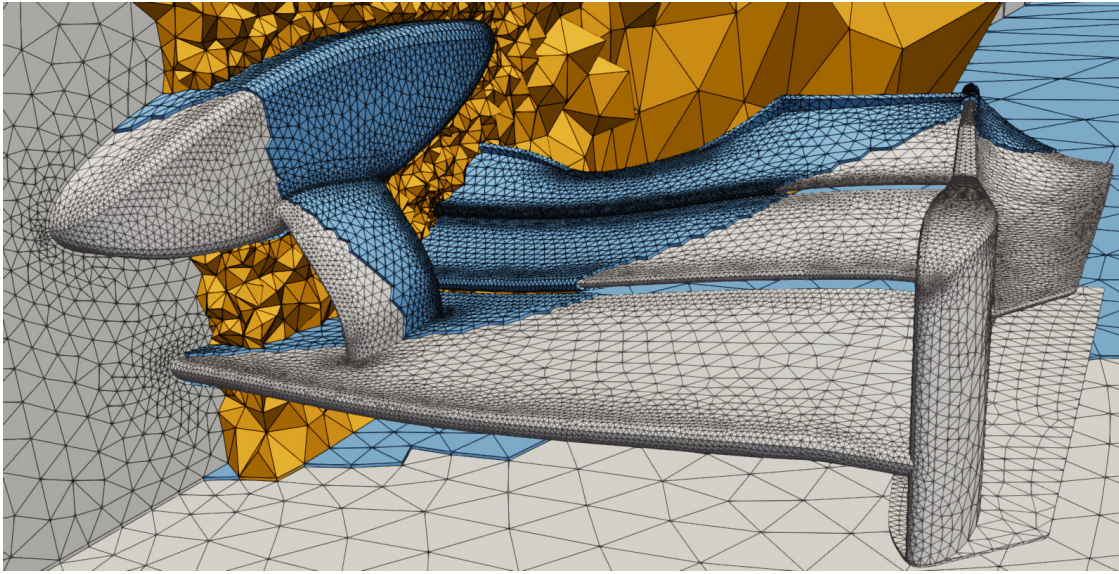


Figure 4.8: A linear mesh of the Imperial Front Wing. The surface mesh is coloured in grey, the prism layers in blue and the tetrahedra in yellow.

Once converted to NekMesh format, the CAD-Reconstruction algorithm is successfully applied as shown in Table 4.3. The application of bounding boxes extending the surfaces facilitated the association of all vertices to at least one BRep object, even if vertex deviation was initially with a maximum deviation $\delta = 3 \times 10^{-6}$ from the BRep. The only exceptions are 5 elements located at the end-plate region where the CAD conformity was deliberately not enforced, leading to multi-patch linear elements. These 5 elements amount for less than 0.01% of the prismatic elements on the boundary. The module did not associate CAD objects to 34 edges, representing 0.04% of the surface edges. Note that the discrepancy between elements without CAD object and edges without CAD objects is expected due to the retrofitting nature of the process and the independent queries of edges and elements that include tolerances.

Table 4.3: Statistics on CAD-Reconstruction module. Vertices, Edges and Faces are only counted on the IFW.

Mesh Bnd Obj.	Overall	No-CAD/ $J < 0$	Fail[%]
Vertices	27 839	0	0%
Edges	83 339	34	0.04%
Faces	55 499	5	0.01%
Elements Tets	398 548	0	0%
Elements Prism	56 592	0	0%

The most critical step for these geometries is the mesh curving to the desired polynomial order, in this case to $p = 5$ with Gauss-Lobatto-Legendre quadrature. As there are also elements and edges without CAD object, the curving module first uses the fail-safe simplified projection on these 34 edges and isolates them from the optimisation processes. For the rest of the boundary edges and elements, we can use the more accurate parametric projection and the surface optimisation presented in Section 3.3.3. This leads to a highly smooth isoparametric surface, see Figure 4.9, even for coarse mesh resolution in regions with high non-uniform curvatures.

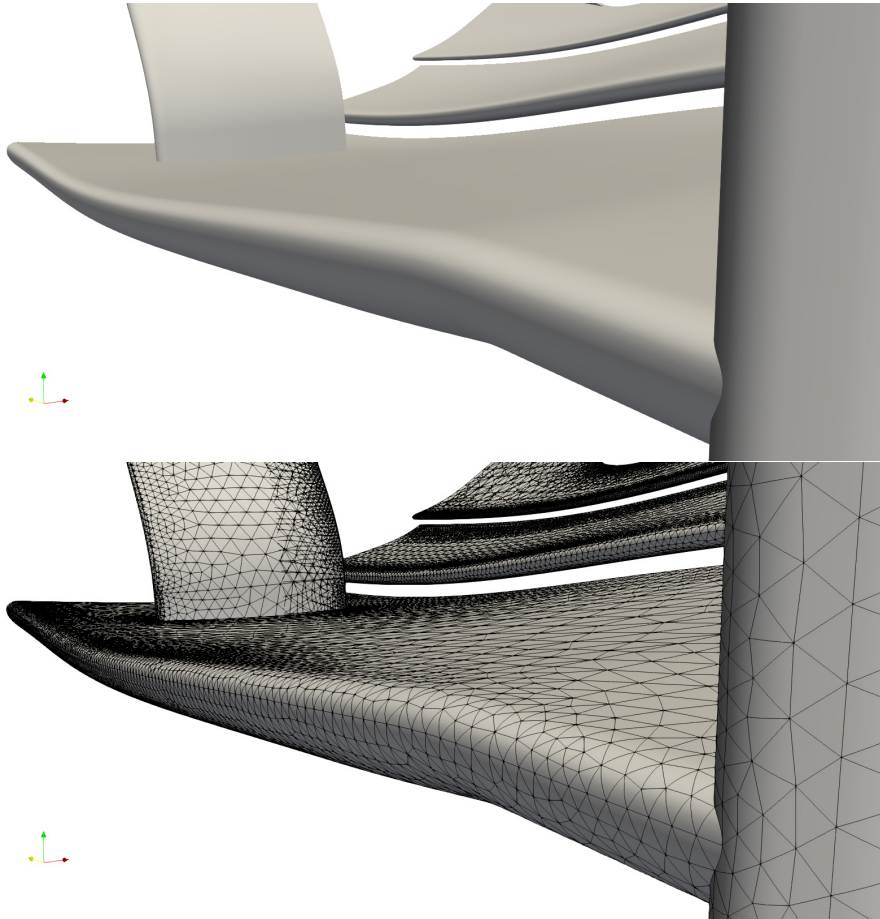


Figure 4.9: Surface quality shown for an enlargement of the isoparametric surface mesh.

As expected, the smooth isoparametric surface from Figure 4.9 results in high geometrical accuracy once it is estimated. The vertices are ensured to be CAD-conforming and lie within a distance $\delta \approx 10^{-10}$, hence within the CAD-API tolerance. The deviation from the CAD is

$\delta = 6.4 \times 10^{-7}$ when tested with $p = 6$ evenly spaced points. The maximum values appear in the regions with the highest non-uniform curvatures and coarsest resolution, with leading edges reconstructed by just two or three faces, and around the simplified projections, see Figure 4.10. The module indicates some higher values around the symmetry plane, but after observation, these were found to be artefacts of the secondary run of face-nodes associations and not real surface discrepancies. Depending on the applications and the Reynolds Number, the user might use this information and assess if mesh resolution or polynomial order should be increased.

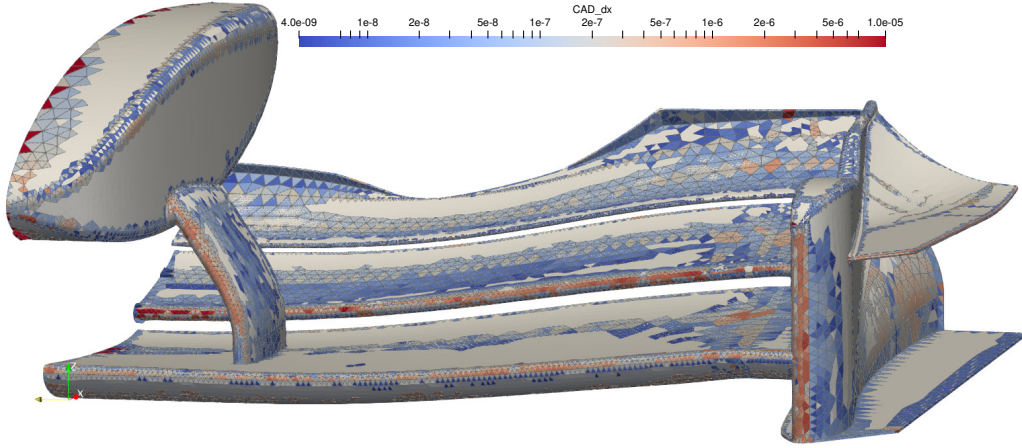


Figure 4.10: The maximum elemental geometrical deviation from the BRep of the isoparametric mesh as evaluated at evenly spaced intervals with $p = 6$. Only surface elements with $\delta > 10^{-9}$ are visualised.

It is also possible to apply the variational optimiser to improve mesh quality. Initially, the mesh is invalid and has five elements with negative Jacobians. The quality after applying the optimiser and linearising the leftover tangled element is displayed in Figure 4.11. The histogram does show improvement in prisms with overall good quality ($J_{sc} > 0.8$), but not as good as the quality obtained for the hemisphere. This can be explained by two factors. Firstly, the initial linear mesh is highly distorted because we enforced CAD compliance despite the presence of very thin CAD sliver surfaces. Secondly, these slivers only allow for very limited CAD sliding in the example, since the surface nodes can move within a CAD surface, but not in between. These issues indicate the need for improvements to CAD sliding between two BRep objects and to high-order mesh modifications. However, such improvements are left for future work.

Finally, following the generation of the isoparametric boundary layer that ensures a suitable mesh resolution for the required local distance y^+ , this workflow provides high-order meshes that are currently employed for incompressible flow simulations.

Finally, we recognize the fact that the proposed meshes will produce an unresolved simulation in almost every region of the domain. However, since we are able to achieve a good geometrical accuracy and overall mesh quality with this coarse approach, we aim at adopting the coarse x^+

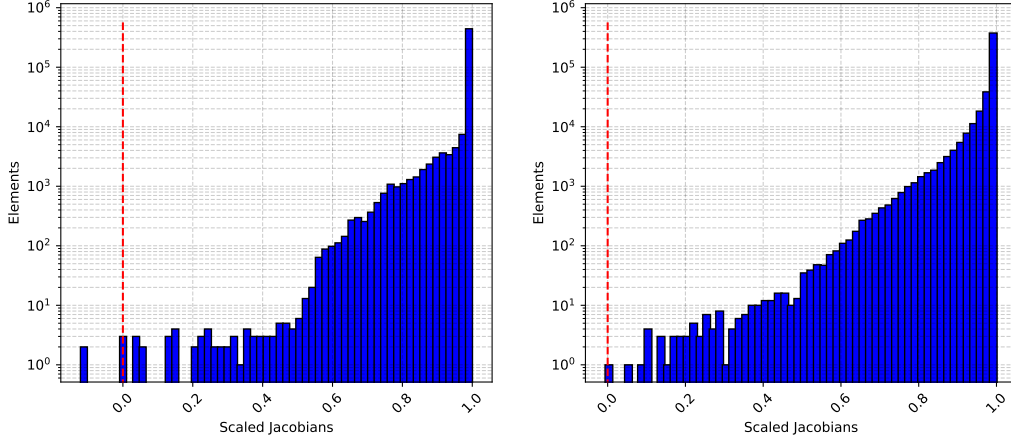


Figure 4.11: Histogram of the distribution of scaled Jacobians in a high-order IFW mesh with $p = 5$ after mesh curving and the end of the meshing workflow.

and z^+ philosophy for the initial high-order meshes and the computational expensive meshing operations. We theorise that, once the user has confirmed correct simulation setup, the user can achieve the desired resolution by uniformly splitting all tetrahedral edges resulting in 1:8 tetrahedral and propagating it to 1:4 prism splitting respectively. This allows the user to test and debug often settings like boundary conditions set-up, time stepping schemes and initialization significantly faster and possibly even simulate the coarse case on a single compute node decreasing queuing time and overall turn-around time for semi-automatic workflows. A more desirable and efficient scenario will be the application of *a posteriori* driven Adaptive Mesh Refinement (AMR).

Industrial Validation By using this algorithm, the Imperial Front Wing was studied extensively with the Nektar++ solver as part of the SSeCOID project. Something more a version of the workflow was implemented by the industrial partner into their CFD process and tens of meshes, part of the IFW, were produced in an automatic way with only several specifically defined meshing macros. This proves the industrial suitability of the workflow for medium complexity geometries. Multiple simulations with different h/p resolutions were performed and the studies results are published by Liosi & Khurana [52], [53] [80], etc., with Liosi [52] showing improved correlation against experiments for the IFW:

"This study demonstrated that high-order spectral/hp methods are a suitable alternative for performing implicit LES simulations for engineering applications at high Reynolds numbers. It is important to model the geometry with increased boundary layer resolution to capture laminar to turbulent transition, especially for industrial geometries whose design point can be sensitive to separation instead of transition.

...

We showed good quantitative agreement with previous experimental surveys by comparing the velocity components between simulation and experiments. The simulated flow structures ap-

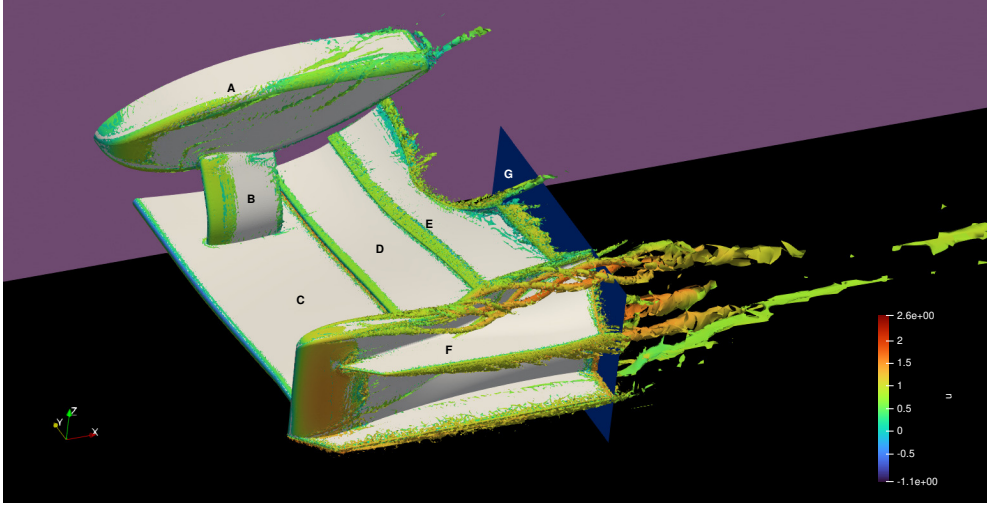


Figure 4.12: Time-averaged $\lambda - 2$ vortical structures on the solution at $Re = 220 \times 10^3$ with the Nektar++ incompressible solver using a mixed formulation with $p = 3$ for pressure and $p = 4$ for velocity. [52]

peared well-defined in the examined position. The simulation and experiments produce similar results for the coherent structures that are present in the experiments. The largest difference is observed below the footplate where a separation region forms in the simulation, which is not clear in the experiments. - A.Liosi [52]

The geometrical accuracy of the produced meshes was deemed to be superior to those produced by the legacy workflow. The simulations produced smooth surface pressure and shear stress τ_{xz} fields that did not suffer from the previously described fishnet patterns thus validating one of the goals of this thesis - high geometrical accuracy for meshes of industrial relevance. The extensive meshing and simulation campaign of the Imperial Front Wing were followed by more complex geometries. The author had the opportunity to apply the workflow on full *R&D* Formula 1 cars of the current(2020-2025) regulations. Three linear meshes were generated and curved up to polynomial order $p = 4$ and one being fully valid and ready for simulations, split prism layers and mesh size of $>30M$ elements. While this was shown to be a major stepping stone towards the goal of industrialization of the method, no simulations were ran due to regulatory limitations. More importantly this was an invaluable experience in identifying the main limitations of this workflow.

4.2.1 Some Limitations And Important Notes of the methodology

First, it is necessary to use multi-patch faces. The enforcement of compliance to a single surface is not practical and leads to sliver mesh faces, see Figure 4.13 a) and then volume ones that decrease the stability of the simulations. Hence, in the majority of the simulations, we use the multipatch elements, see Figure 4.13 b). There is a minimal deterioration of the

geometrical accuracy, but the overall stability of the simulations was improved substantially and no high-skewness elements are detected.

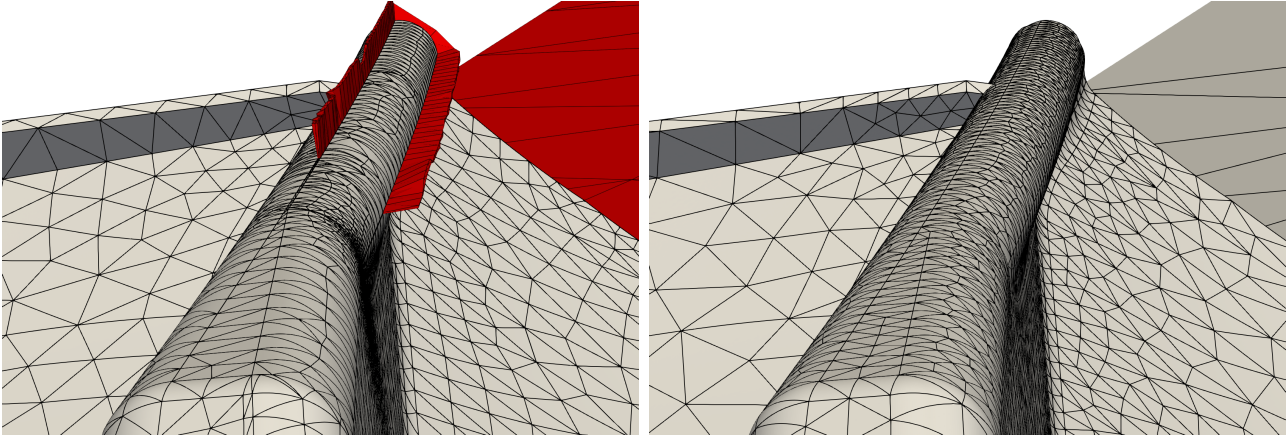


Figure 4.13: The top of the IFW end plate where a CAD sliver has generated linear elements with high skewness (a), which are easy to tangle during mesh curving or volume optimization. The use of multi-patch elements (b) in the region eliminates the skewed elements.

For the more complex geometries than the IFW, extensive manual iterations and remeshing in the third-party software were necessary until a suitable linear mesh is generated to facilitate the mesh curving process, please refer back to Figure 4.14. When working with new very complex geometries where the x^+ and z^+ resolution requirements are not known a priori and very high multi-directional curvature is present, we often observed small but crucial details of the mesh to be tangled or to have low quality elements due to coarse linear mesh, see Figure 4.15 a,b). A straightforward and effective strategy is to enforce a sufficiently high curvature refinement on the third-party software while not enforcing a minimum surface size, see Figure 4.15c).

While successfully resolving the issue, it does increase the overall mesh count in other, less important regions, and does require the manual interaction, taking often on the order of hours, or *a priori* detailed knowledge. All this has been drastically worsened by the strict O-type prism layer, which sometimes necessitates coarser elements to prevent the formation of pyramid patches. In the industrial context, one can also rely on the extensive experience and *a priori* knowledge. However, this contradicts some of the initial goals we set for a general-purpose workflow aiming at diverse applications and complexities.

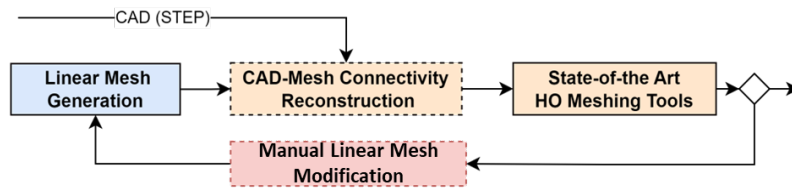
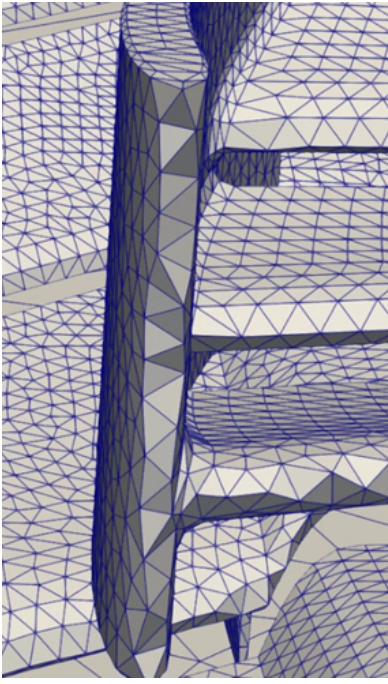


Figure 4.14: The mesh generation workflow in reality for more complex than the IFW geometries. In red the new manual interaction in the feedback loop.



(a) Initial Linear Mesh

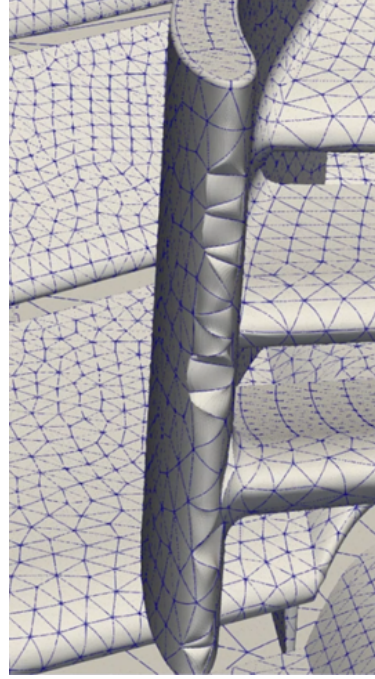
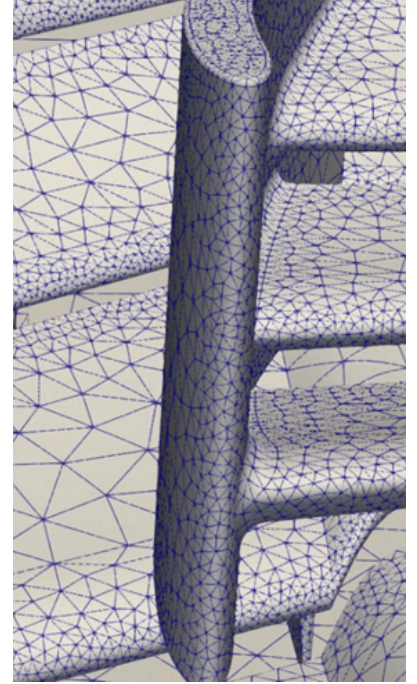
(b) $p=4$ ($J<0$) curved in NekMesh from the initial linear mesh(c) Final $p=4$ ($J>0$) mesh after remeshing in StarCCM+ and re-curving in NekMesh

Figure 4.15: An automotive example, where an initially coarse linear (a) mesh resulted in inaccurate mesh curving with tangled elements (b). Final valid mesh after multiple remeshing attempts in StarCCM+ and applying our pipeline (c). Please note the increased curvature refinement and extra DoFs on the other leading edges, where the initial mesh was actually of acceptable mesh quality even with the initial linear mesh.

Last but most importantly, the restrictive nature of the isoparametric boundary layer splitting that does not support generic boundary layer meshes but splitting only O-type macro prism layers. This is not always possible to generate, as in some regions of the geometries, such as cotangent surfaces. In these regions, it is natural for one of the prismatic edges to collapse, often generating a small region with pyramid and tetrahedral elements, see Figure 4.16. This forces the third-party software or the user to decrease the minimum allowed mesh quality and minimum allowed prism layer thickness, see Figure 4.13 a) and Figure 4.16 b) respectively. As the isoparametric prism layer splitting by Moxey [73] assumes constant number of prisms around the whole body, accounting for $> 40\%$ of the overall mesh count, this could also become a restriction from the CFL perspective and the overall simulation cost.

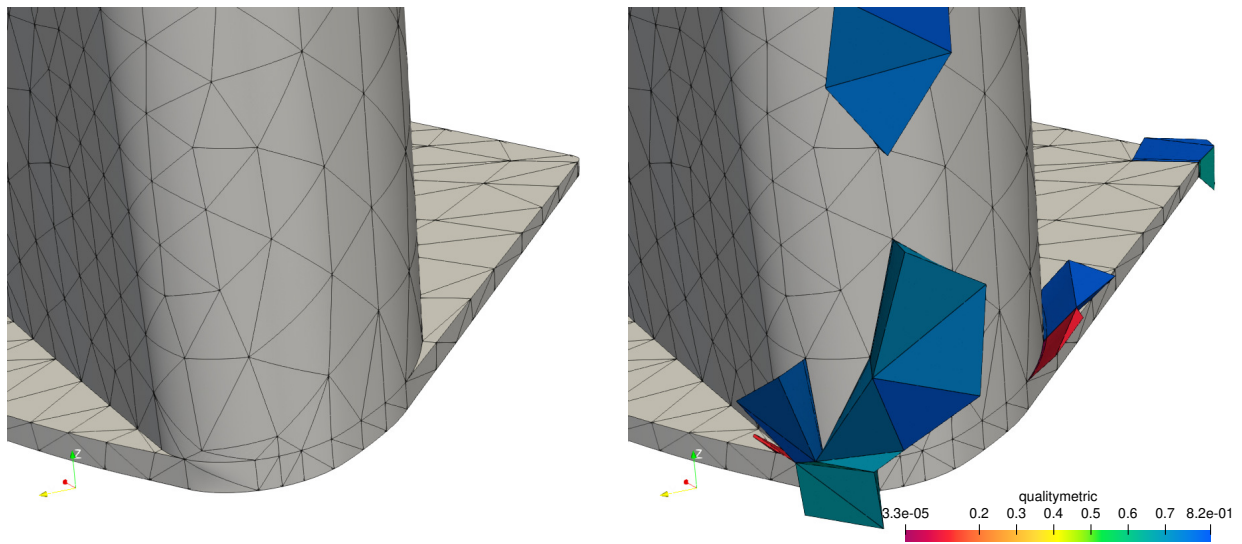


Figure 4.16: The IFW cotangent region on the footplate with a fully prismatic macro layer, where the cotangencies have forced the generation of low quality mesh slivers, which are further deteriorating after the mesh curving (red).

Chapter 5

Advanced Meshing Concepts

All previous applications have demonstrated the consistency, flexibility, and robustness of the proposed high-order mesh generation workflow when applied to third-party linear meshes. However, we also described some of the general *a posteriori* limitations and the ones of this implementation. Recognizing their importance, this work aims at alleviating and if possible solving these through mesh modifications. A proof-of-concept of the novel solutions are presented through two industrial examples - an automotive case Perrinn Front Wing [85] and the NASA CRM High Lift Prediction Workshop (HLPW) Case 3.

5.1 Closing the loop of Automation

We try to address the need for extensive manual iterations and remeshing in the third-party software until a suitable linear mesh is generated to facilitate the mesh curving process. Creating a perfect mesh from the first attempt is near impossible and multiple iterations are usually necessary in both $x+$ and $y+$ directions both *a priori* and *a posteriori*. This work proposes the use of high-order mesh modification techniques within **NekMesh** as an alleviation of this issue.

High-Order Isoparametric h-refinement Both 2D and 3D high-order h-adaptation techniques were implemented following a split, swap, smooth process in 2D and a split, smooth process in 3D. The main part - the high-order splitting happens in the reference space (ξ) of the element, based on predefined maps and on the edges marked to be split, for example 1:8 tetrahedra, see Figure 5.1. The implementation supports both high-order curved tetrahedra and prismatic elements. Additionally, a propagation algorithm ensures conformity even with a subset of all maps, in our cases >130 Maps, see Figure 5.2. This way, by creating the new elements in the references space of the parent element, the high-order curved adapted elements

keep the geometrical accuracy and the mesh quality of the parent element. For more details, please refer to Part II of this thesis, where these techniques are described more thoroughly.

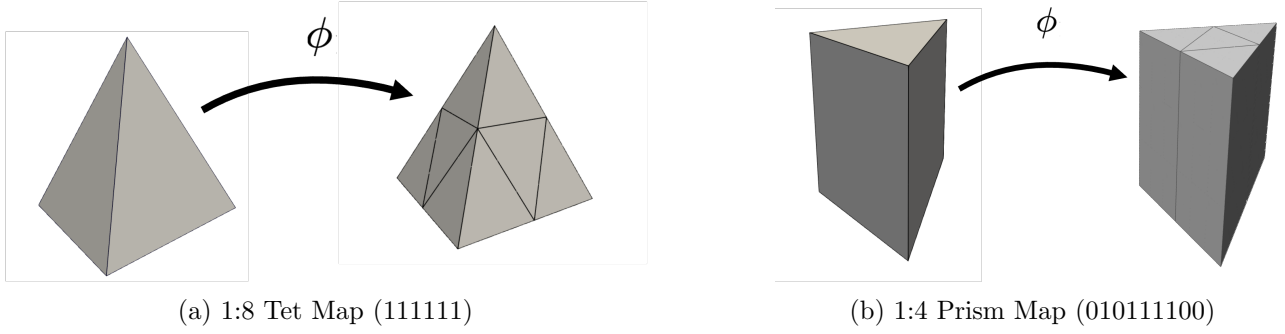


Figure 5.1: High-order refinement, where the split happens with a predefined map in reference space.

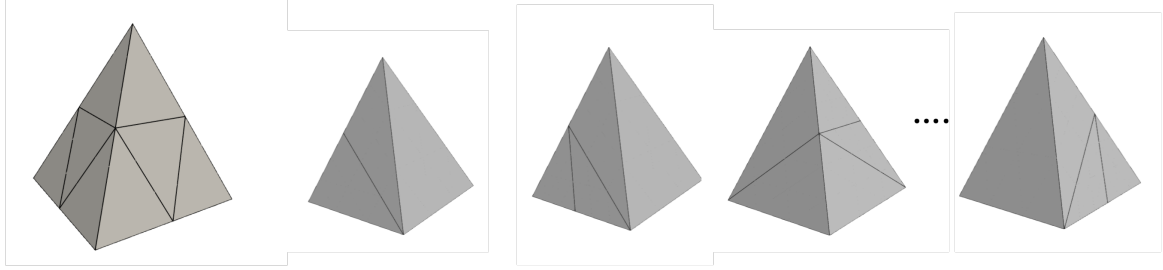


Figure 5.2: High-order refinement, a small part of the >130 alternative maps in reference space to ensure conformity between split.

Automation - The Algorithm The process works as follows: Given a linear straight sided mesh, we first apply the mesh curving (projection + optimizations, but not prism layer splitting) up to the polynomial order of choice as per our proposed workflow. Then we calculate the elemental mesh quality, based on the scaled Jacobian J_{sc} and shortlist the elements with $J_{sc} < 0.6$. Although the precise threshold can be examined further, targeting the most distorted elements is consistent with Khurana’s [80] findings that these elements often govern the Δt . We re-curve only these selected elements to p2. Then we apply the high-order splitting operator on the selected low quality elements with a 1:4 pattern for the prisms, changing the basis function to quadratic (with the mid-points of the edges projected onto the CAD). Since our h-refinement is conformal, this is sufficient and it will propagate the refinement to neighboring prismatic and tetrahedral elements lying on top of the prisms. Once these are refined, we check the validity of the underlying linear element, and if valid we apply the mesh curving process to the polynomial order of choice only for those refined elements.

Algorithm 2 Closing the loop of automation

```

1: Input: Coarse Unsuitable Linear mesh  $\mathcal{M}$ 
2: Output: High-Order Refined mesh  $\mathcal{M}_{ho}$ 
3: procedure PROCESSMESH( $\mathcal{M}, p$ )
4:   Reconstruct Mesh-CAD link
5:   Curve  $\mathcal{M}$  to  $p$ 
6:   Compute  $J_{sc}$ 
7:   Shortlist  $\mathcal{M}_{el} \leftarrow \{el \in \mathcal{M} \cap J_{sc} < 0.6\}$ 
8:   for all  $el \in \mathcal{M}_{el}$  do
9:     Curve  $el$  to  $p = 2$ 
10:    Split  $el$  1:4 prisms with  $p = 2$ 
11:    Ensure mid-edges on the CAD
12:   end for
13:
14:   Propagate the conformity
15:   Split conformally prisms and tets with  $p$ -desired
16:
17:   for all  $el \in \mathcal{M}_{el}$  do
18:     if  $el_{p1}$  then
19:       Curve  $el$  to desired  $p$ 
20:     end if
21:   end for
22: end procedure

```

Perrinn Front Wing We select the Perrinn Front Wing as the next test case - an open-source full F1 car designed to be compliant with the 2015 regulations [85]. We first isolate the front wing from the full car and make it watertight with the help of ITI CADfix [39]. This geometry is a step forward in terms of geometrical complexity, containing all parts of the IFW, but also a four-element wing for generating Y250 vortex, an upper cascade with turning vanes and strakes under the mainplane.

The linear mesh is generated with StarCCM+ with a single thick layer around the body and the floor similar to the IFW coarse mesh, see Figure 5.3. It is curved and optimized as per the third-party workflow to $p = 4$. At this point, we observe that we have achieved overall good surface representation and geometrical accuracy, see Figure 5.4.

However, one can also observe the significant deterioration of the mesh quality due to mesh curving on the leading edges(LE) of the Turning Vane and its elements, see Figure 5.5 b). This is a common use case, which appears in regions with very high curvature but low mesh resolution, resulting in invalid or very low geometrical accuracy. Opposite to the IFW, where some of the LE CAD Surfaces were split at the point of maximum curvature, here a single CAD Surface spans the whole turning vane. This leads to edge-nodes that are either too far from the CAD parametric location or their curving tangles the element.

At this point, the proposed mesh modification process is applied - calculating the full mesh J_{sc} , changing the $J_{sc} < 0.6$ elemental basis to quadratic and ensuring these $p2$ edge-nodes

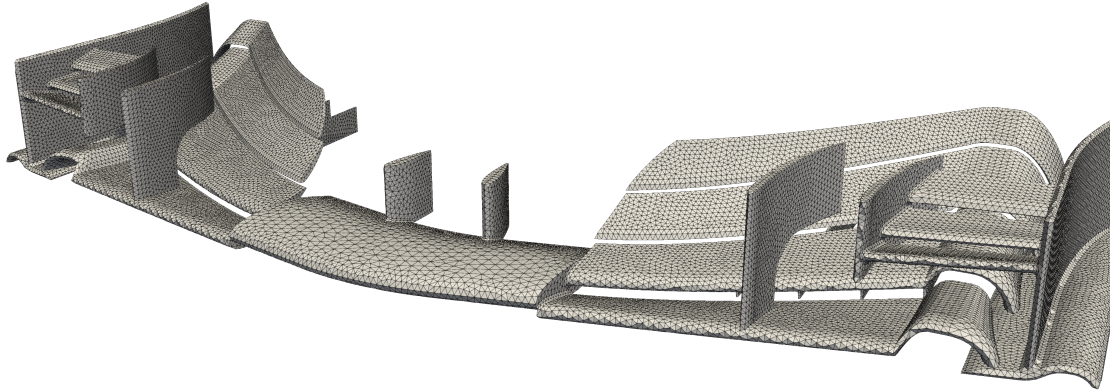


Figure 5.3: A coarse Linear Mesh for the Perrin Front Wing

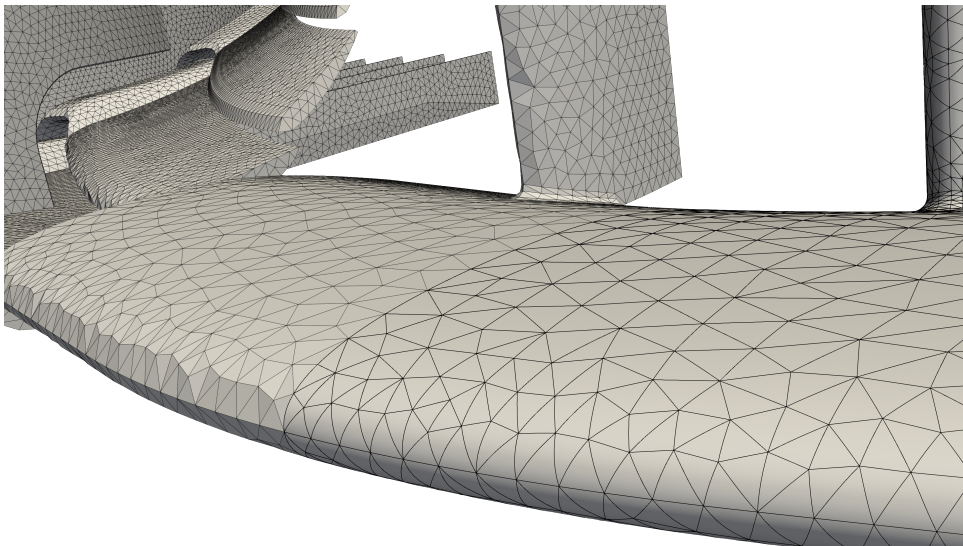


Figure 5.4: A comparison of a coarse region between straight sided and high-order $p=4$ surface mesh of the Perrinn Front Wing

are one the CAD. While this mesh does not represent the geometry well, it does suffer from fewer very low-quality ($J < 0.05$) elements - 209 vs 102. Hence, this mesh represents a better choice for the isoparametric splitting that is applied to the valid curved elements. At this stage, it is possible to split these valid prisms with a 1:4 conformally. Lastly, we curve the selected elements to the initially desired polynomial order $p = 4$. The final refined high-order mesh can be characterized as notably improved, with less distortions and with clearly improved smoothness and subsequently geometrical accuracy, see Figure 5.5 c).

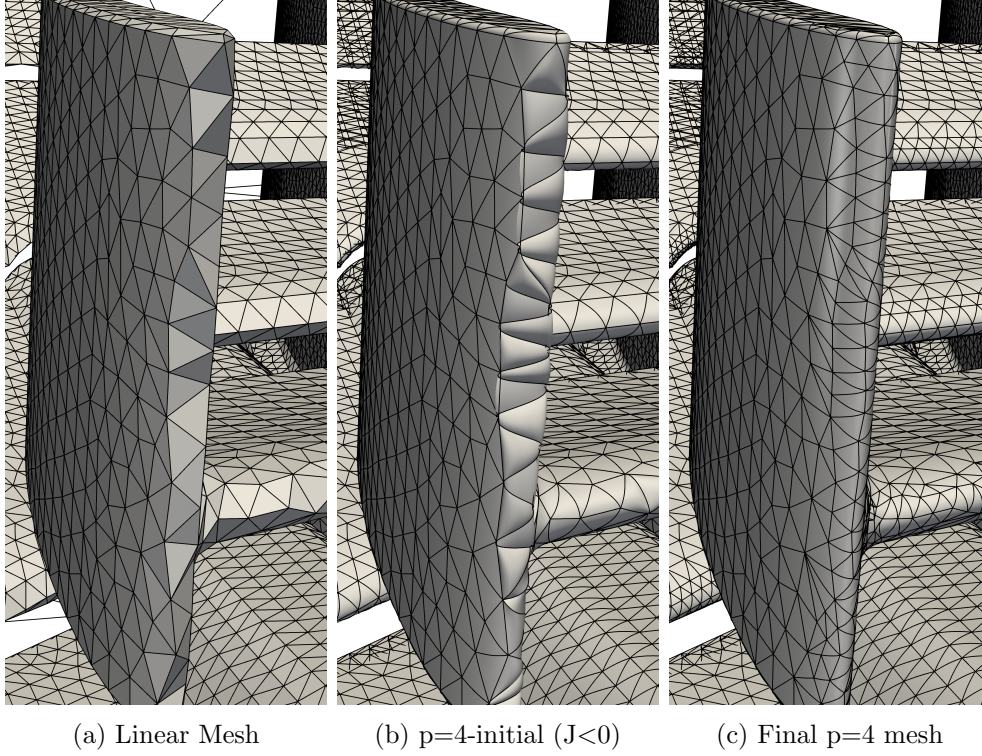


Figure 5.5: High-order hp-mesh refinement fully performed in NekMesh for mesh generation. Note that the global resolution of the mesh is preserved and only the distorted elements on the leading edges are refined.

Looking at the volume mesh, the hp-refinement we applied naturally leads to an increased mesh count, but only by 7%. It is also possible to validate this improved mesh quality by comparing it before and after the process, see Figure 5.6 in the region of the turning vanes. Initially, there are 44 invalid elements on the vane. After the process the mesh quality is drastically improved, 25 of them being untangled even before the use of the Variational Optimizer. Another visible improvement is the endplate, where the number of lower quality elements ($J_{sc} < 0.6$) decrease from 512 to 213 (58% reduction).

This approach can be combined with the Variational Optimizer for approximative mesh curving only for the last problematic elements. This would enable a type of closure of the manual feedback loop previously reliant on third-party tools, allowing in some cases the entire workflow to be contained within NekMesh automatically once the initial unsuitable for mesh curving linear mesh is generated.

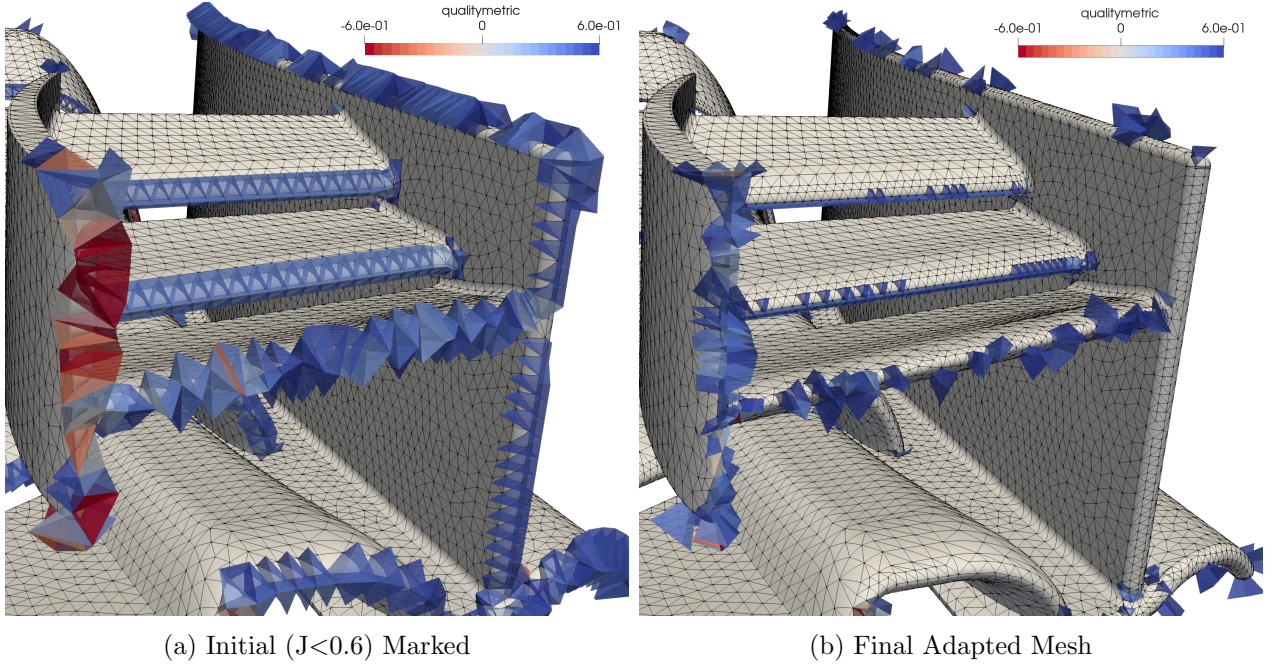


Figure 5.6: Extraction of the highly distorted elements $J_{sc} < 0.6$ for refinement, before and after proposed process of hp-refinement.

5.2 Generic Viscous Layers

Last, we attempt to alleviate the decreased mesh quality due to the restrictive use of O-type macro prism and the lack of support for generic boundary layer that contain also pyramids and tetrahedra. As one of the main contributions of this thesis, we propose the application of our high-order mesh modification techniques to reformulate the long-standing isoparametric prism layer splitting to a high-order conformal refinement. This new process allows meshing much more complex geometries and improves the mesh quality of the initial linear meshes.

In the standard isoparametric prism layer approach, considered in the Section 3.5, the macro prism is split in the wall normal direction locally in a single shot with the same uniform progression ratio and a constant number of layers across the entire body. While this approach might be feasible for O-type prism layers, it becomes impractical in the presence of mixed element types such as prisms, pyramids, and tetrahedra, as it would necessitate a large set of splitting maps to maintain conformity, three new ones for every new progression ratio or even for a new set of number of layers.

Instead we opt for a gradual high-order refinement, where first all prisms are split, this forces the conformal propagation algorithm to split also the pyramids with 1:2. This naturally means introducing two new splitting maps for the pyramids, see Figure 5.7. Splitting the triangular faces, however triggers the splitting for other pyramids or tetrahedra, see Figure 5.8. And since all tetrahedral maps are already introduced, this is naturally handled by our new mesh modification module, guaranteeing full conformity. It is important to acknowledge that while

solving the third-party workflow largest limitation, if split to a large number of layers (e.g. in high Re cases) and without further improvement, this leads to a high valence on the top of the pyramid and the undesired highly skewed pyramids.

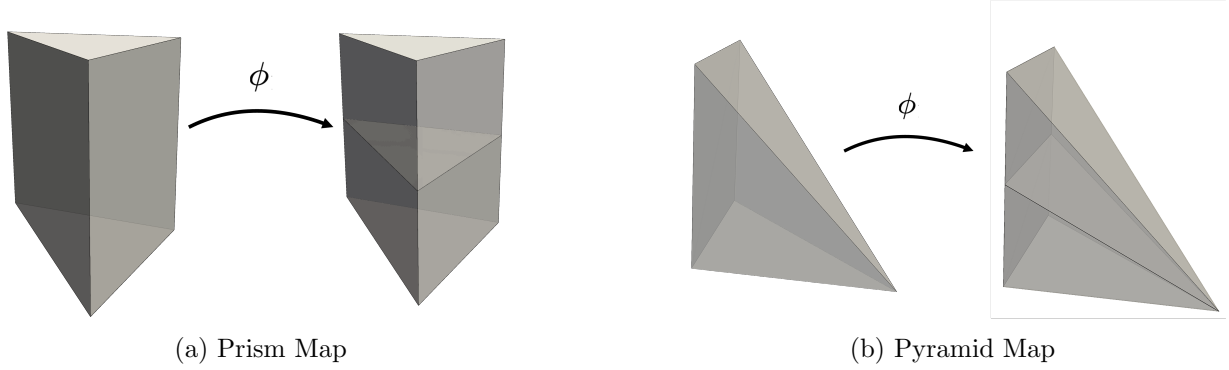


Figure 5.7: Some of the Additional Prism and Pyramid Maps added

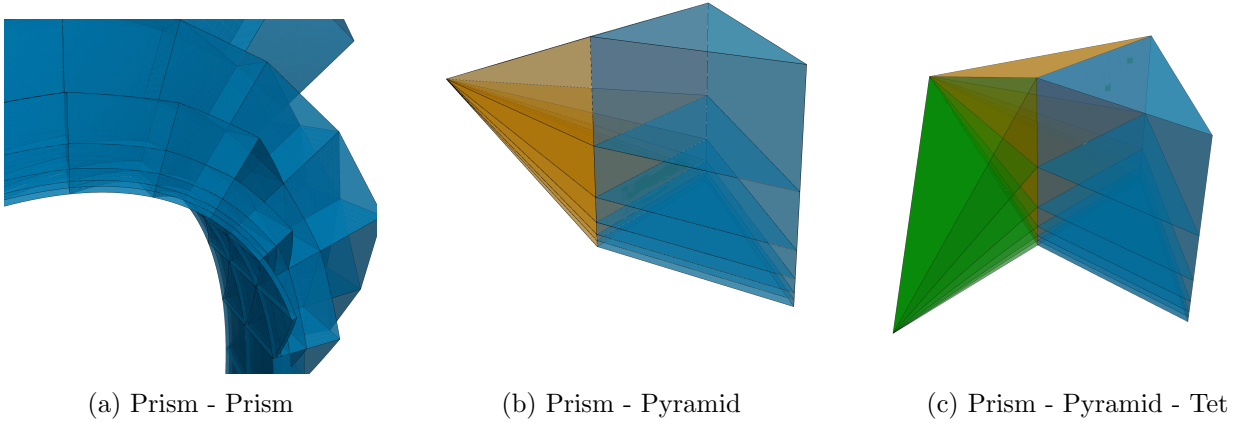


Figure 5.8: High-order refinement combinations used for boundary layer mesh generation

We consider two additional treatments of this isoparametric splitting to fully obtain both efficient generic high-order viscous layer splitting but also good final mesh quality. First, the gradual refinement strategy allows the generation of variable number of layers in the different regions of the geometry, which also could alleviate the skewness of the pyramids as fewer layers can be generated there. More importantly, this also presents a major efficiency and stability improvement, as the number of elements is expected to decrease by up to 30% for complex geometries. The number of prism layers can be driven based on AMR or by geometry in the proximity regions where the macro prism layer has collapsed. This requires the introduction of a series of splitting maps for the prism to 1:3 (2 pyramids + 1 tetrahedral element). Moreover, with another three prism maps 1:4 and an extra map for the pyramid, we can also improve the dihedral angles of the pyramid elements. This will be beneficial as poor angles and high apex valence can increase the condition number of the elemental matrices and possibly degrade near wall accuracy. These last two functionalities, however, are not yet fully implemented in our proof-of-concept, due to the time limitations of the thesis and some Nektar++ specific vertex ordering requirements for the collapsed coordinate system.

The Fifth High-Lift Prediction Workshop - CASE3 As a final example showcasing the proof-of-concept of our generic boundary layer splitting, we have selected the full aircraft geometry of the Fifth High Lift Prediction Workshop Case 3 with extended flaps and slats, see Figure 5.9. We chose this geometry due to the difference in the characteristic lengthscale (Mean Aerodynamic Chord 7m and fuselage length 62m) and high-curvature concave and convex regions. Moreover, it does contain cotangent regions, see Figure 5.10, specifically around the leading edge slats. These have been previously reported as challenging to mesh curve with fully tetrahedral elements as part of the HLPW4 by Ruiz-Gironés and Roca [93].

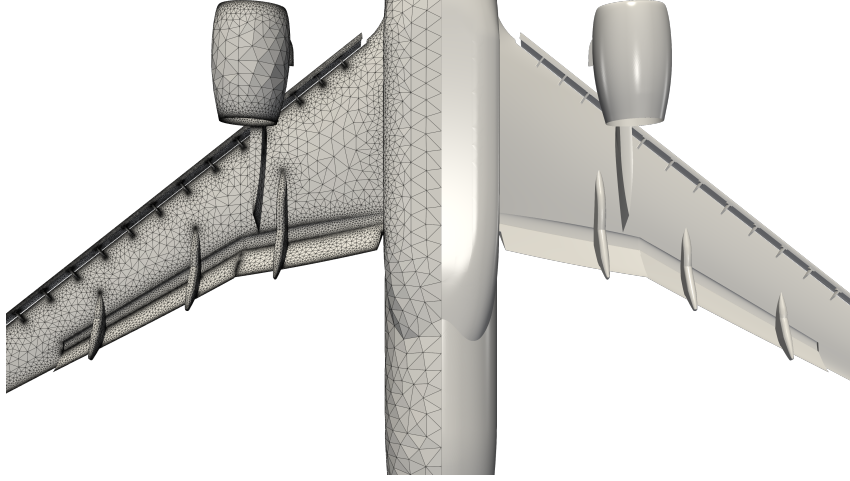


Figure 5.9: Linear mesh on the left and the high-order surface mesh on the right showing the complexity of the high-lift flap-slat configuration.

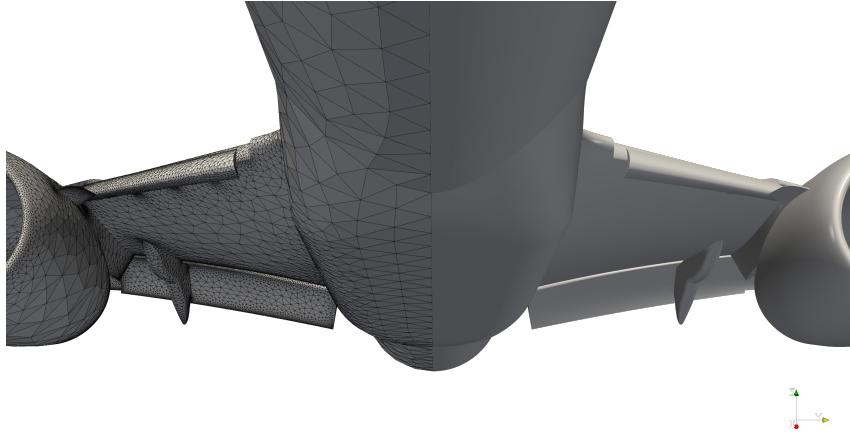


Figure 5.10: Linear mesh on the left and the high-order surface mesh on the right. Note the cotangencies on the slat-mainplane regions.

We start with a linear mesh generation in a third-party commercial software, attempting to achieve a hybrid fully O-type prism layer and core tetrahedral mesh. However, in this case, regardless the allowed deterioration of the prism quality, this is not possible and the tool generates 66 pyramids, grouped in 12 regions, see Figure 5.11 a). Some of these correspond exactly to cotangent regions reported previously. Another challenging part of the geometry is the connectivity between the flaps and slats pylons with the mainplane. In these regions,

there is a junction point, where it is topologically impossible to define a single normal direction that will be suitable for generating a prism normal outside of the body and that would be "visible" from all surfaces and their prisms. We would refer this as a multinormal regions and an example can be seen in Figure 5.13 a), where 5 CAD Surfaces with normals pointing opposite to each other prevent the generation of an O-type of prism layer. Hence, the linear mesher will inevitably fit a group of pyramids to fill this region.

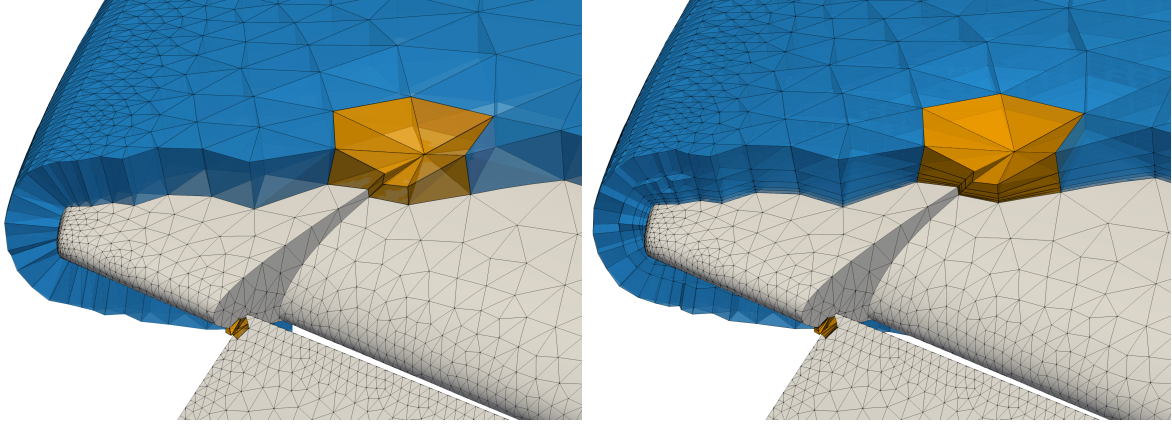


Figure 5.11: HO Mesh Pyramids regions corresponding to regions of cotangent surfaces before and after the adaptation.

Once in the NekMesh environment, we use the proposed pipeline for CAD-reconstruction, projection and surface optimization for mesh curving. No volume optimization was applied due to implementation limitations on the Variational Optimizer in NekMesh for pyramids. The resulting mesh is of good geometrical accuracy with 70 invalid elements, none of which are in the regions of the pyramids. After linearising the invalid elements, the mesh is valid and we can proceed with the viscous layer splitting.

This case is suitable for our generic viscous layer splitting. As a proof-of-concept we apply the high-order h-refinement four times, always targeting the surface elements, which results in five prism layers with aggressive progression ratio of 2.0, with the bottom two elements having the same height.

Mesh Object	HO Before	$J < 0$	HO After	$J < 0$
Elements	1.46M	70	2.5M	0
Tetrahedra	1.2M	0	1.2M	0
Prism	260k	70	1.3M	0
Pyramids	66	0	330	0

Table 5.1: Summary of the HO-mesh before and after the isoparametric split. The invalid elements of the initial HO-mesh were linearised before the split.

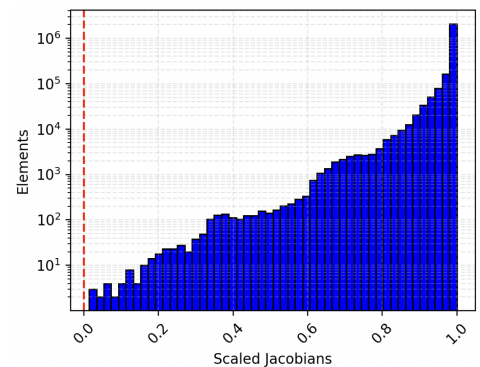


Figure 5.12: Mesh quality in terms of J_{sc} for the final HO-split mesh.

In the cotangent region of the leading edge, see Figure 5.11b, this results in near identical first cell height as the rest of the prism layer. In the multi-normal regions, see Figure 5.13b, one can observe a collapse of the macro prism due to still forcing the the linear mesh generator for minimum use of pyramids. This once again suggests the potential of the approach, not only allowing meshing more complex geometries, but also being beneficial for the solver, since one can collapse the macro prism layer to pyramids earlier and hence keep the overall macro layer more uniform.

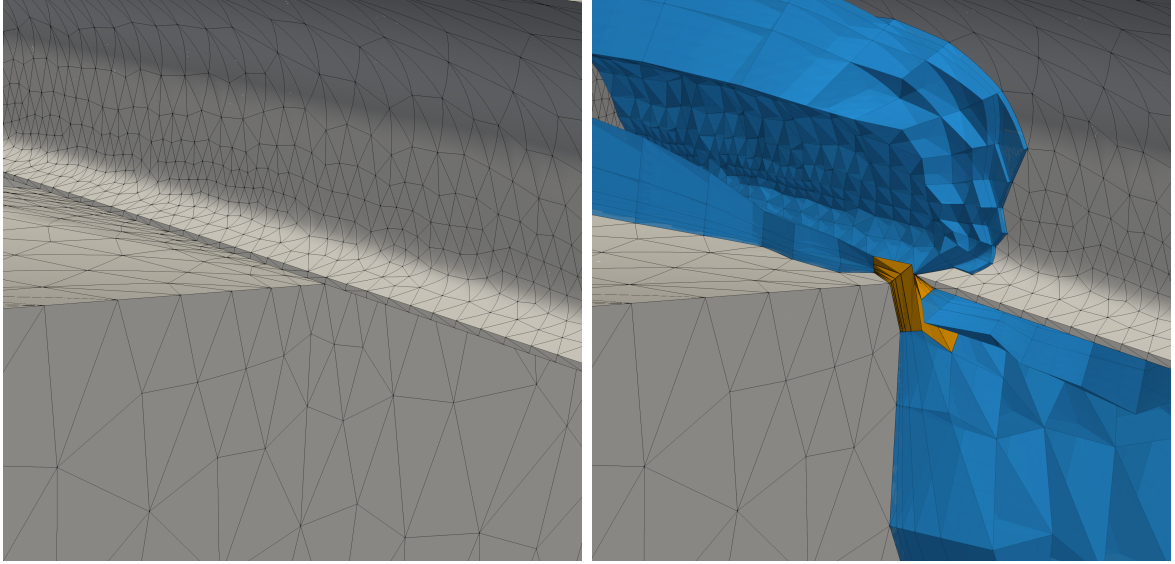


Figure 5.13: Multi-normal region with 5 surfaces having opposing normals making it unsuitable for O-type of prism layer generation. Hence a pyramid region - surface mesh (a) and after the splitting (b).

All in all this final mesh is valid and conformal, passing the NekMesh connectivity test. To validate its validity and conformity we also interpolate the mesh successfully to an uniform flow with Nektar++. No full simulations have been run for this case due to the computational cost of the model.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

As a step towards the industrialization of high-order CFD methods, this work has focused on providing a robust *a posteriori* mesh generation pipeline. We attempted to develop a generic end-to-end process that produces isoparametric viscous meshes with high geometrical accuracy and good mesh quality, while targeting extreme geometrical complexity and seamless integration with already existing automatic industrial CFD pipelines.

Starting with a detailed analysis of some of the limitations and challenges of *a posteriori* high-order meshing and with the mentioned objectives in mind, this thesis proposed a new workflow for high-order mesh generation from third-party meshes. A central contribution to this work and the backbone of the pipeline is the novel method for reconstructing the link between the mesh objects and the underlying BRep one, the CAD-Reconstruction. The process is completely generic with minimal limitations, agnostic to where the mesh comes from and the only assumption is that the mesh vertices are reasonably close to the CAD. This offers several advantages over other third-party techniques found in the literature.

First, it allowed us to seamlessly reuse and improve some of the NekMesh high-order mesh curving, surface and volume mesh optimizations into a full end-to-end high-order mesh generation pipeline with various types and sources of linear mesh. Starting from some academic test cases, we managed to show a high degree of **consistency** between our bottom-up workflow - both from a geometrical accuracy and mesh quality perspective, even for very coarse elements. The new workflow curves the linear mesh, producing the same geometrical accuracy, and takes advantage of both the high-order surface, but also the volume variation optimizer of NekMesh with sliding the surface vertices and high-order nodes on the CAD.

Second, the reconstruction module is agnostic to the third-party linear mesh generation and can be applied to any state-of-the-art modern high-order mesh curving and optimization techniques that use CAD parametrization. This means there is full **flexibility** coming from the CAD-Reconstruction module in terms of standard finite-element mesh types. We present this in terms of three different mesh sources of the extruded Imperial Front Wing and the three different meshes in 3D (hexahedral/tetrahedral/hybrid tet-prismatic). All three generated coarse curved meshes with a smooth surface and overall good mesh quality.

Moreover, the new pipeline enables users to leverage multiple sources of CAD truth. This significantly reduces the requirements on the CAD BRep for mesh curving and leaves the existing linear mesh generation process unchanged - a major advantage from a robustness perspective. While not the focus of our work, there were several additional capabilities that were obtained by extending the CAD-reconstruction to high-order meshes: assessing geometrical accuracy, optimizing and modifying both internal and third-party high-order meshes a posteriori with CAD parametrization knowledge.

Going back to industrialization, we presented the results of a comprehensive simulation campaign of the Imperial Front Wing and its configurations. The third-party methodology was used for the extruded and the full Imperial Front Wing within an industrial partner's CFD process and hundreds of highly stretched valid high-order hybrid meshes were produced automatically after some initial effort. All featuring significantly improved geometrical accuracy, even for extremely coarse meshes. The simulation campaign by Liosi and Khurana showed an improved correlation with other numerical and experimental data. This was partially attributed to the improved geometrical accuracy of the proposed industrial meshing workflow, which did not suffer from previously seen geometrically driven non-physical pressure and shear-stress artefacts. Hence, validating the method in terms of geometrical accuracy and the possibility of achieving y^+ resolution.

This success was followed by a gradual increase of the geometrical complexity, reaching an F1 car without CAD simplifications or repair. This case represented a major milestone in validating the proposed industrial pipeline since the mesh-CAD link reconstruction, the mesh curving and the surface optimization were applied for a very coarse high-order prism-tet mesh. However, this more complex case also showed the limitations of the methods. First, there was an excessive need for manual interventions to produce a coarse, straight-sided mesh suitable for curving mesh. The stumbling block was the thick O-type macro prism layer that is required by the original isoparametric prism layer splitting. In certain topological regions, such as cotangent surfaces and proximities, the macro layers were found to collapse up to 2% of the desired thickness, drastically deteriorating the mesh quality and producing pyramids that can be connected to other type of elements. This makes the original isoparametric splitting concept inapplicable.

As a step further, we implemented a novel for NekMesh high-order isoparametric h -refinement, specifically designed to operate on the curved geometries that also propagates the splitting for conformity. All this, while keeping the geometrical accuracy and overall mesh quality of the initially valid mesh and without costly CAD operations. To address the workflow limitations, we presented two proof-of-concept solutions with complex industrial geometries. First, a potential for automatically modifying the initial third-party mesh, creating a suitable mesh for mesh curving a posteriori. This can be seen as an $x+$ resolution control performed at a lower quadratic order for better robustness of the distorted elements, followed by an additional local mesh curving to the desired order of interest in those regions. We present an F1 application, where this was found to increase the overall quality and even virtually untangle invalid elements for very coarse curved meshes in high curvature regions. This is expected to bring even further benefits once integrated with the variational optimiser.

The final and possibly the most impactful contribution of this thesis is the reformulation of the isoparametric prism layer splitting from a single shot requiring an O-type mesh with a constant thickness setting to a progressive multi-step high-order conformal h -refinement. This permits the viscous mesh to contain prisms, but also tetrahedral and pyramidal elements. The initial proof-of-concept of this approach was positive, allowing meshing to previously impossible geometries, such as the HLPW5, both maintaining the elemental curvature, but also propagating it inside the thick prism-pyramid-tet boundary layer mesh without the need for an additional mesh untangling or optimization.

While some further work is necessary for precise thickness and variable prism-layer progression control, the proposed work represents one of the last steps towards the fully automated high-order workflow we propose for high geometrical complexity, flexibility and seamless integration with existing CFD workflows, making it not only possible but also delivering in a "fast, robust and consistent" manner.

6.2 Current & Future Work

While the author believes the thesis has increased the Technology Readiness Level of high-order methods, the work will not be done before the industry switches to the tool. Hence, we highlight several directions of future work that the author and the NekMesh team are currently interested in.

Mesh Curving, Variational Optimizer and Untangling While the Variational optimizer has not been the focus of this thesis, we identify two crucial areas for improvement - the underlying assumption being that the CAD is valid without slivers and last but not least, the need for a fail-safe mechanism for CAD-sliding. The assumption is highly unlikely to be satisfied, with gaps, slivers, etc. always present. Currently, the optimizer does not possess the capability for sliding a node between CAD surfaces, limiting the effect on these elements and in case of a sliding outside of the CAD surface or a surface parametrization being invalid, the method will abort the optimization. For tackling the industrial geometries, one could envision two possible solutions.

First, after a problematic iteration or one that moves a node outside of the available uv-parametric space, one can try to use the CAD-reconstruction to update the CAD-connectivity of these elements and if necessary, project them back to the new CAD surface. This will be a quick solution but will still rely on some sort of CAD validity. A more robust, but also more involved, approach would be the reparametrization of the underlying CAD. This could take advantage of the intrinsic C^0 continuity and the connectivity of the curved surface mesh - either directly sliding on the coarsest surface mesh or on a more sophisticated Bezier approximation, for example, borrowed from the IGA community.

Variable Polynomial Order Curving The NekMesh software architecture allows seemingly easy extension of the mesh curving modules to introduce a variable polynomial. This can be achieved by employing the geometrical accuracy metric explained in Section 3.3.3 and the fact that the faces and edges shared between elements are unique and are not duplicated. Therefore, no smoothing or optimization is necessary for enforcing variable polynomial order on the final isoparametric element used for the solver. Our initial 2D experiments showed a potential of linearizing up to 10% in the viscous mesh without noticeable deterioration of the geometrical accuracy, but also need to curve up to p5 for very coarse elements with non-uniform curvature. Since solvers treat "deformed" and "straight-sided" elements differently, this has the potential to decrease the overall simulation cost.

Fail-Safe Automation Finally, for the last few elements that the variational optimizer is not capable of untangling and we usually linearize in the end, it is possible to attempt local h-

refinement on the straight-sided element, integrated within the optimizer and follow with an approximative mesh curving. This can be further improved if combined with local p-continuation on these elements, starting from a quadratic basis for robustness. With this extra step, the author expects to close the automation loop completely within NekMesh.

Generic Variable Viscous Layer One of the most impactful part of this thesis has been the high-order h-refinement and the reformulation of the isoparametric prism layer splitting to a gradual h-refinement that can be performed on any type of elements (prism, tet, hex, pyramid) and without necessarily going for a full O-type of viscous layer. While the majority of the work is completed, there are still some minor additional functionalities required. First, in the HLPW5 example, we did split gradually the elements 1:2 in the middle ($\xi = 0.0$) of the reference element. Nothing stops one from selecting ξ at the desired location to achieve a different than 1:2 progression ratio. This is a simple, high-priority development.

As a final step for the viscous layer and a benefit of using the isoparametric gradual h-refinement is the possibility for a variable number of prism layers meshing across the body. This is also one of the main recommendations, which is currently in development and requires only three extra "transitional" maps for the prism splitting. Given our extensive experience with >130 available maps, it is straightforward and the only extension is creating a controller that defines what regions of the prism layer need to be adapted at every h-refinement iteration. However, the effect of this minor change is expected to be a significant efficiency and stability gain on the solver side. It is expected to allow up to 30% decrease in the overall mesh count and could provide a ground for more effective *a posteriori* AMR applications tracking the exact y^+ requirements. This should solve all remaining challenges we identified previously and will result in a high-quality generic viscous layer meshing even for very complex geometries such as full F1 cars or aircraft in high-lift configurations.

High-Order Edge Collapse To complete the mesh modification work, it is recommended to implement a high-order edge collapse in 3D. The work on this has not been started, but a potential high-order edge collapse that preserves the geometrical accuracy of the mesh will have a profound impact on solving the slivers presented on the IFW and always present in industrial geometries. Moreover, this will make the surface meshing easier and possibly without the need for Case 3 multi-patch elements, making the CAD-reconstruction and the mesh curving more robust as the quality can be controlled through topology *a posteriori*. This is expected to improve CFL and simulation stability.

Performance Through this thesis, the author has worked with a wide variety of mesh sizes, with the largest reaching up to 40M elements. While all presented cases were curved or refined

in a matter of minutes, the time complexity of the methods is not linear and hence the largest mesh could take hours. This can be attributed to the serial nature of all NekMesh modules except the variational optimizer. Moreover, NekMesh uses a separate legacy data structure from the core Nektar++ and often rebuilds or destroys the Nektar++ objects, meaning a large memory footprint, with hundreds of GB of memory requirement for the largest tests and even more for the variational optimizer.

Hence, all modules developed in this thesis have been implemented to be straightforward for OpenMP or MPI parallelization in mind. Moreover, there is already an effort to migrate NekMesh to a lightweight version of the Nektar++ geometrical objects, which have shown promise for fixing both the memory and the algorithmic inefficiencies of NekMesh. Last, this will automatically grant access to the extensive performance optimizations, collections and MPI-based infrastructure of Nektar++, making the subsequent parallelism of NekMesh easier.

Part II

Adaptive Mesh Refinement

Chapter 7

Literature Review & Research Questions

7.1 Motivation

Once it is proven that an initial coarse high-order mesh with good geometrical accuracy and quality can be constructed for the geometries of interest, the focus shifts to how one can correctly and efficiently capture the flow features in scale-resolving turbulent simulations as cheaply as possible. In an ideal world, one can envision no computational limitations, where Direct Numerical Simulations with uniformly fine meshes reaching Kolmogorov length scales without a need for resolution control are used. This is unlikely to be the case for any complex 3D industrial ones, where the Re often reaches $\mathcal{O}(10^7)$ and is impractical even for a full exascale supercomputer. Hence, it is natural to concentrate the available resolution only in the regions of interest and specifically for given computational resources. Consequently, the aim is to resolve the large eddies while modelling the dissipation from the smaller ones as in the Large Eddy Simulation (LES), but as it is often seen in practice, we might even leave these unresolved and perform underresolved LES simulations that are sufficient for answering particular engineering problems. The research interest of this project is how and where these degrees of freedom (DoF) are allocated. The hp/SEM allow us to explore the full spectrum of control: standard mesh size control, such as splitting elements and moving meshes, but also variable polynomial order across the domain and within the individual element.

Starting from the incompressible flow applications, the presented IFW already showcases the main challenges and is a good test case for the required robustness and efficiency even for the initial mesh generation. For the IFW it is essential to resolve the exact transition mechanism and its location on the surface. However, the industry application also demands resolving down-

stream vortical structures, which are later used for flow control purposes. As will be explained further in the thesis, this might require a different approach for defining the h/p resolution than for the compressible cases. Moreover, the high Re and required wall shear stress with separation & transition locations are also limiting factors, but this time on the computational efficiency for resolving the boundary layer and, hence, of the whole CFD simulation pipeline. An example iLES on this geometry with the solver of interest, Nektar++, is shown in Figure 7.1 for $Re=2.2 \times 10^5$ manually meshed with about 3×10^6 elements and polynomial expansion order $p=4,5$, translating to an overall cost of $\mathcal{O}(10^7)$ DoF [52]. The combination of the sheer number of DoF and the very small explicit timesteps $\Delta t = 1 \times 10^{-6}s$ becomes the computational bottleneck for iLES with industrial geometries.

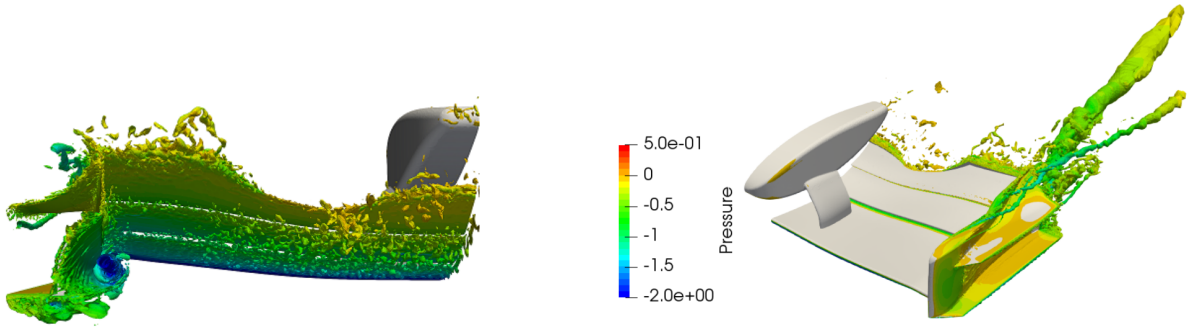


Figure 7.1: λ_2 iso-contours coloured by pressure iLES of the IFW by Buscariolo et al. [11]

On the other hand, the simulation of an isolated supersonic high-pressure turbine poses the most of the challenges present in compressible flows. The geometrical complexity for the initial mesh generation is minimal, considering the single extruded airfoil. However, due to the supersonic regime, a shock forms in front of or on the airfoil, as shown in Figure 7.2. These features with very high gradients and discontinuities could cause numerical oscillations and divergence of the solver, either the Euler or the Compressible Navier-Stokes. Therefore, they will need to be identified through shock-capturing methods and handled differently. These features can sometimes be treated as steady-state flows, both in steady and unsteady solvers. Most often, however, for high angle of attack (AoA) or high Re, the airfoils develop detached flow regions, large recirculation areas such as separation bubbles, vortex shedding and shear layer instabilities. These are not only hard to detect and resolve, but are also intrinsically unsteady and are advected downstream. This can also destabilize the shocks, making them an even harder case for adaptive mesh refinement (AMR). The combination of shocks and discontinuous flow phenomena with the turbulence and vortical structures found in incompressible flows make this even more complex to solve.

Such quasi-steady cases have been widely used to test the robustness of AMR mesh approaches with 2D airfoils [49, 4, 60, 27]. Most of these were only focusing on 2D steady problems, with only beginning at the end of this PhD project, the state-of-the-art hp/SEM slowly transitioned to 3D unsteady cases as in the work by Massaro [78] and even first dynamic DG-FVM as by

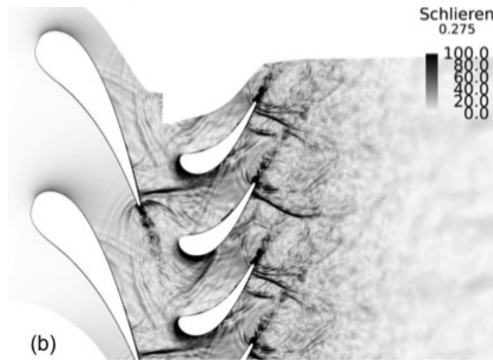


Figure 7.2: Density gradient of a high-pressure turbine stage [81].

FLEXI [68]. All these, however, limited to simpler geometries.

All this being said, the purpose of this research is to identify an efficient adaptation strategy, not to establish the optimal one, and employ the specific hp/SEM properties in the right combinations. Moreover, to improve the overall computational cost, the level of automation, and achieve high-order AMR applicable for steady and unsteady industrial flows. In particular, AMR for high-order methods with curved meshes. All this should also be applicable to complex industrial geometries. Even a hypothetical reduction of 20 times in DoF could translate to a transition from the one national Tier 1 HPC system to a Tier 2 or even Tier 3 HPC system, allowing a wider industrialization of high-order methods and more efficient utilization of CFD.

7.2 Mesh Adaptation Overview

Mesh adaptation or adaptive mesh refinement (AMR) is a process of controlling the discretization error of PDEs by controlling the number, the location or the type of DoF in a computational domain. AMR can be *a-priori*, without any knowledge of the solution, mainly during the mesh generation. Alternatively, *a-posteriori* AMR when the mesh is optimized using information from the computed solution. The focus of this work will be on *a-posteriori* methods only.

Generally, AMR methods share the same general structure. First, an initial sub-optimal mesh is generated and a solution is calculated with the solver of interest. Based on this solution, an error is estimated and the associated mesh metric field corresponding to the desired resolution through the domain is calculated. To achieve this target resolution, the mesh is adapted to the solution by means of topological and geometrical modifications, the previous solution is interpolated to the new mesh and the solver is restarted. This cycle is repeated until a user-defined stopping criterion, usually related to the error metric or maximum computational cost, is reached. This could be interpreted as a closed-loop control system, as shown in Figure 7.3.

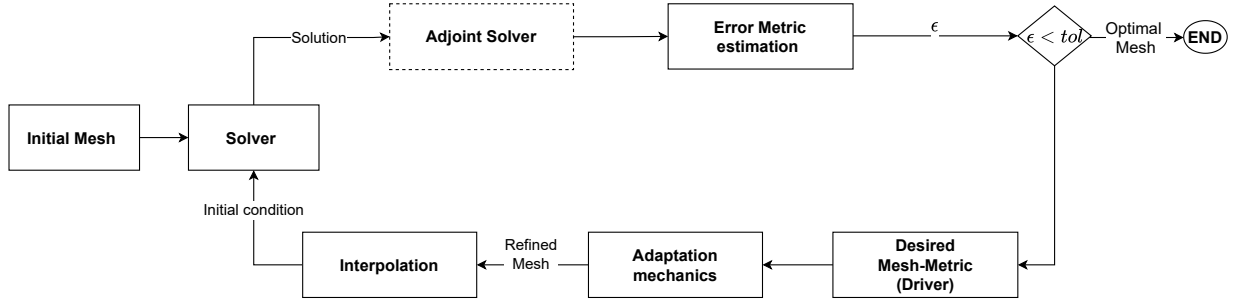


Figure 7.3: Control loop of mesh adaptation

7.3 CFD Solver

As in Part I, Part II focuses on high-order numerical methods. Over time, they have evolved to a variety of discretization methods, for example, the Continuous Galerkin (CG), Discontinuous Galerkin (DG) and the Flux Reconstruction (FR), which can be considered a generalization of DG methods. The project aims to develop a modular AMR strategy that remains independent of any specific solver implementation and valuable to the whole high-order community. Hence, it is essential to account for the discretization differences and similarities in the context of AMR and their suitability for the applications.

The primary computational framework used in this project is the Spectral/hp Element Method (SEM) framework Nektar++. For a detailed description of the solver, the reader is referred to Karniadakis and Sherwin [43], [19] and [70]. In Nektar++, both Continuous Galerkin (CG) and Discontinuous Galerkin (DG) SEM solvers are used for incompressible and compressible N-S solvers. Hence, we will focus the discussion on these two discretizations, how they handle different flows features and the interfaces between elements, so the proposed AMR will be compatible with both.

7.3.1 CG Discretization - Incompressible Navier-Stokes

In the classical Continuous Galerkin discretization, the nodal expansion (DoF) is shared on elemental interfaces. This way the solution is forced to be C^0 continuous between elements, see Figure 7.4 a). This leads to beneficial properties for smooth flow, such as the spectral (exponential) convergence. Moreover, CG has by definition less DoF compared to the DG, as no extra DoF are introduced into the system compared to the DG discretization. [43]

However, since DoF on the interfaces are shared, the global assembly of the mass matrix becomes highly coupled. Therefore, the method is less likely to be applied for hyperbolic problems and compressible flows with discontinuities like shocks. In Nektar++, a CG formulation is used for the Incompressible N-S solver. [71]

From a mesh adaptation perspective, the shared DoF has significant consequences. It means

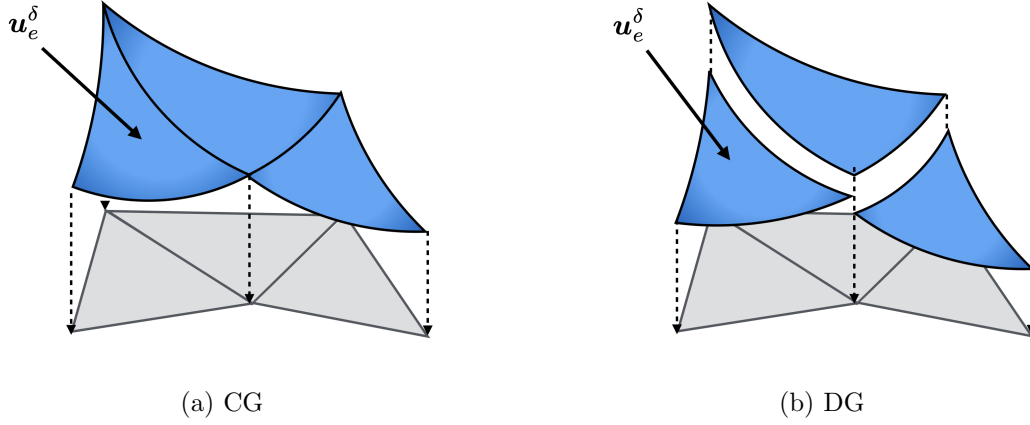


Figure 7.4: Comparison of the solutions between the CG and DG discretization. Please note the C^0 continuity of the CG formulation and the jump in the solution for the DG.[26]

that any local mesh modification or a local change in the polynomial order of an element directly affects its neighbours. Moreover, the global conformity of the whole nodal space on the elemental interface should be enforced at any point of the simulation - non-conformal meshes are not naturally allowed for CG, requiring special treatment. Last, the global matrices need to be assembled again after every mesh modification.

7.3.2 DG Discretization - Compressible Navier-Stokes

The DG formulation, on the other hand, does allow for the discontinuities on the solution by design, see Figure 7.4 b). More importantly, each element in the DG discretisation has its own basis functions and DoFs, which are independent of those in neighboring elements. The communication between elements is performed through the elemental fluxes in the weak formulation of the Governing Equation, which are separately defined in the elemental traces, usually forming a block diagonal global stiffness matrix [43].

This decoupling makes them the prime discretization tool for hyperbolic problems with discontinuities and, consequently, compressible flows. Naturally, they have a higher number of DoF than CG, but are also easier to scale on modern architectures.

What is more important for this part of the thesis is their natural alignment with AMR. They can handle variable polynomial orders even without fully rebuilding the global matrices. Last, almost all high-order solvers using non-conformal meshes are based on DG discretization - MFEM/Blast, Flexi and Horses3D to name the few [30] [3] [68]. The compressible flow solvers of Nektar++ use full DG formulations. In the current implementation, both the CG and DG Nektar++ solvers require conformal meshes.

7.4 Error Indicators and Mesh Metrics

The second step of the AMR control loop is estimating the error and the desired resolution through the domain. While this is out of the scope of this work, it is likely the most comprehensively studied part of AMR. Therefore, only the main definitions and conclusions will be given here and the reader will be referred to reviews by the Unstructured Grid Adaptation Working Group (UGAWG) [4], the ones by INRIA Gamma3 team [1] and Vivarelli [109] primarily focused on lower order methods, and last but not least, the high-order reviews by Fidkowski [31] for adjoint and the comparison of p-type error indicators [49].

7.4.1 Feature-based Error Indicators

The standard feature-based error indicators are arguably the most widely used today. They identify a flow region with a specific feature assuming large error will be present around these regions, but not directly approximating the discretization errors. Having such a generic definition allows versatility, but a sensor working for one application is not suitable for all flow regimes or features.

For example, high gradient or the second derivative Hessian, L^q -norm, of a flow variable is the most common way to guide AMR for hyperbolic flows with discontinuities like shocks.

Another similar group of feature sensors, particularly useful for high-order methods, are **smoothness sensors**. In the presence of discontinuities with an element, one can expect to see an increase in oscillatory behaviour in higher frequency modes (Gibbs phenomenon). The first such local indicator for DG SEM was proposed by Persson and Peraire [86], where the energy of the highest DG mode is used as an indicator.

Naturally, one can also rely on the **flow features** such as Q-criterion, λ_2 for targeting specifically vortical structures or the dilation and the DG stabilization terms for shock capturing. While all these indicators are straightforward for implementation and suitable for any solver, one must acknowledge the main drawback being one or more parameters and thresholds that have to be tuned for the particular problem at hand and the major assumption that adapting only the features is sufficient to improve the results.

7.4.2 Truncation Error estimators

The local truncation error, on the other hand, can be defined as the difference between a discrete operator and its continuous counterpart, both evaluated with the exact solution.[88] In the AMR context, this could be the higher terms of the polynomial expansion of the basis function.

The most common indicator for smooth flows and CG SEM was proposed by Mavripilis [63], which is based on the modal energy decay of spectral basis functions compared with the ideal exponential decay. This discrepancy is then used to estimate the local truncation error.

Some efforts in the use of such indicators were made by Kompenhans et al. [49]. They advocate that since the truncation error is the local source of a discretization error through the discretization error transport equation, hence it is fundamentally the preferred choice. Moreover, their numerical tests showed a solution can be achieved with less than half of the DoF of Hessian feature-based ones for p-adaptation. This, however, is true only for smooth flows without discontinuities and relies entirely on being in the asymptotic resolution range and choosing the correct error thresholds for the application of interest.

In summary, since the first attempts for local feature and truncation error indicators for SEM by Persson&Peraire and Mavripilis respectively, many other derivatives have been proposed, for example, a comparison with other truncation-based ones for p-adaptation by Carvalho [18] in the CGSEM context - this work is currently in the process of implementation in the Nektar++ framework. However, as Carvalho describes, the truncation ones are more sensitive to the polynomial order, size and flow feature, which hinders their use for hyperbolic problems.

7.4.3 Goal Oriented Error Estimation

In the last group, the adjoint or goal-based error indicator estimates the sensitivity of the target functional (e.g. lift, drag, mass flow) with respect to a perturbation of the flow somewhere in the domain. In other words, the adjoint identifies the areas of the domain where the local flow has the greatest impact on the final target functional [1]. A notable work in this area is the one by Fidkowski [31]. To do that, an additional Adjoint solver needs to be implemented for the system of PDEs and then to be solved after the main solver iteration, marching backwards in time. This approach has been found to converge significantly faster to fewer DoF and in fewer iterations than the Hessian-based approaches [4]. As the goal-oriented adjoint adaptation actually incorporates the physics behind the PDE, it clusters the DoF around where it matters the most, whereas the feature-based ones generally refine the whole domain, see Figure 7.5. The cost of running a second solver, however, also shows the need for a case-specific trade-off on using either adjoint or feature-based error indicators.

7.4.4 Metrics and Adaptation Drivers

Once the error is approximated, the mesh is optimized by a mesh modification operator that is controlled by a so-called driver. In the anisotropic FEM case, this usually translates to a metric tensor transformation accounting for the size scaling, rotation, aspect ratio, and skewness from

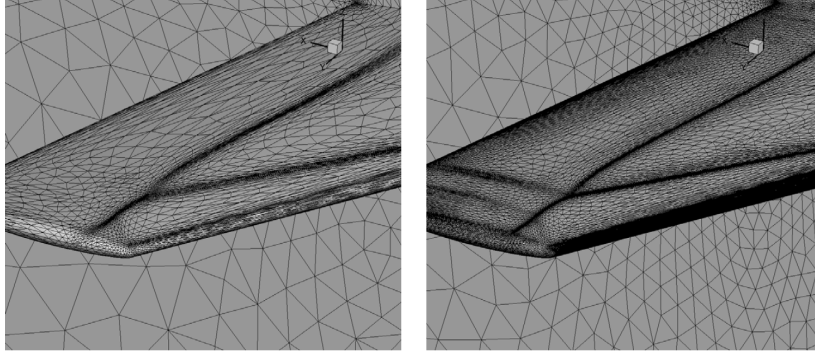


Figure 7.5: Adapted mesh ONERA M6 wing: [L^2 -norm (left), adjoint (right)] [4].

the initial to the target mesh $\mathcal{M}$. In the most widely used continuous mesh formulation for interpolation error indicators, this is represented by a field of local Riemannian metric tensors, primarily developed by the INRIA group [55], [56] and followed in the majority of previous studies also by Dapogny, Dolejsi, etc. [23], [17]. Then, by transforming the metric to the desired error and using the error equidistribution principle, one can derive a new local metric in which the eigenvalues and eigenvectors are used to define the local target optimal size, shape and orientation.

As an equivalence to the Riemannian metric approaches, one can formulate it as a metric tensor as described by Peraire and Peiro [83]. To tackle the cost, a dynamic AMR has been achieved in scale through the Target Matrix Optimization Paradigm (TMOP) [47] in the discrete form within MFEM. This approach allowed the LLNL team to fuse multiple mesh quality with Hessian-based error indicators for exascale applications [107]. A standard approach based on the Hessian ($\mathcal{H}$), can be written the following way:

$$\mathcal{H} = Q^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} Q. \quad (7.1)$$

with Q being the orthogonal eigenvector matrix corresponding to the principal directions, and λ_i - the eigenvalues representing the magnitude of the second derivatives along the eigenvectors. This can then be written as the anisotropic mesh metric $\mathcal{M}$:

$$\mathcal{M} = Q^{-1} \begin{pmatrix} |\lambda_1| & 0 \\ 0 & |\lambda_2| \end{pmatrix} Q, \quad (7.2)$$

The error estimator can link the metric to the target element size by using the eigenvalues λ_i , giving the desired size h_i for the respective principal directions. Moreover, the orientation can be defined as a rotation with angle θ coming from the eigenvectors matrix Q , which completes the anisotropic target element definition. A simpler transformation can be defined for isotropic refinement, as done by Marcon [60] in the Nektar++ framework, see Equation 7.4.

$$\mathcal{M} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \frac{1}{h_1^2} & \\ & \frac{1}{h_2^2} \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (7.3)$$

$$\mathcal{M} = \begin{bmatrix} \frac{1}{h^2} & \\ & \frac{1}{h^2} \end{bmatrix} \quad (7.4)$$

7.4.5 Novel Intelligent Error Indicators and Adaptation Drivers

During the span of this PhD project, there was some effort in using the machine learning techniques for estimating the error or defining the density metric. Starting from the work by Lock, who showed that based on extensive industrial aeronautical data, one can train an artificial Neural Network (NN). This NN approximates an improved isotropic or anisotropic mesh spacing ρ_ξ of unseen aircraft configurations through point sources that are then fed to a mesh generator in *a priori* way [54].

From the *a posteriori* AMR perspective, one can mention the work by Huergo and the Horses3D team on the Reinforcement Learning error estimation approach that bypasses the manual tuning of feature-based error indicators with offline training [37]. Another one is the work by Zhang and the Firedrake team [112] on the reformulation of the Monge-Ampere PDE-based mesh movement framework to a transformer-based one. This accelerates both the flow feature detection and the dynamic mesh movement driver drastically in 2D canonical flows. Last, and possibly the most advanced use of agent-based RL algorithm DynAMO by an LNL group, where hp-adaptation was accelerated, and the algorithm predicted flow features dynamically, refining the region of interest even before hyperbolic fronts appeared there, allowing for more efficient AMR [25].

This kind of approach for error estimation and driving the mesh modification techniques are likely the future of AMR, and the author would like to explore them in the future as part of dynamic hp-adaptation for Nektar++. This, however, is out of the scope of this project.

7.5 Adaptation Mechanics and Mesh Modifications

Once the target element resolution h_i is calculated and the zones for mesh refinement or coarsening are identified, it is necessary to discuss the exact refinement operators for this resolution control, which is actually the focus of this thesis. In literature, it is generally accepted to distinguish three main types of mesh modification mechanisms: h-, r-, and p-. Additionally, one may be considered the full remeshing of the domain as an additional -m type or a subset of h-adaptation.

p-adaptation

P-adaptation controls the error through varying the local polynomial order (p) of the spatial discretization.

This approach has several advantages. First, it is relatively simple to implement, especially when isotropic refinement is concerned. The local expansion order is usually changed iteratively by one up to a stopping criterion. This refinement also enables unsteady simulations without requiring costly mesh modifications, using both p -refinement and p -coarsening. Moreover, this can be extended to anisotropic p -adaptation based on the different principle directions of the element as in Horses3D [49] for tensor product structured meshes. Last, this approach for p -refinement has been suggested to be the most efficient for targeting smooth flows and less effective for the ones with discontinuities. The main implementation challenges in p -adaptation are ensuring conservation across interfaces with variable p -elements, load balancing and efficiently rebuilding local and global matrix-based operators.

In DG-SEM, variable p is straightforward since the solution is C^0 -discontinuous and the elements communicate only through the DG-fluxes and the shared trace. Naturally, one can use the higher order for the trace flux calculations, which would mean projecting/interpolating the lower order solution to the high-order space. Then, when the solution on the lower order is evaluated, the flux can be projected back to the desired orthogonal basis. Naturally, the flux can be used as it is for the higher p element. Such approaches have been the most widely adopted for p -adaptation, as in Nektar++, HORSE3D, FLEXI, etc. [27, 30, 67]

The CG treatment, however, is more complex due to sharing the unique DoF on the interface between adjacent elements. Hence, fewer such examples are present for example Deal.II and Nektar++, where the hierarchical basis functions are used to deactivate the high-order modes on the elemental interface to match the shared DoF as per the work by Ekelschot [27].

r-adaptation

Mesh movement or r -adaptation is a powerful approach for controlling the spatial h -resolution through clustering the nodes and consequently deforming the elements without introducing or removing any DoF, see Figure 7.6. This is usually achieved either through a Laplacian smoothing, elasticity based approaches where the energy of deformation is minimized through an iterative optimization of a physical element to a target one with properties (e.g. scale, aspect ratio, skewness, rotation/orientation) corresponding to a metric tensor. Last, it is possible to use moving meshes by solving a PDE approach such as the optimal transport method of the Monge-Ampere PDE as proposed by Budd [10].

R -adaptation has been found to improve the quality of the solution with both feature-based

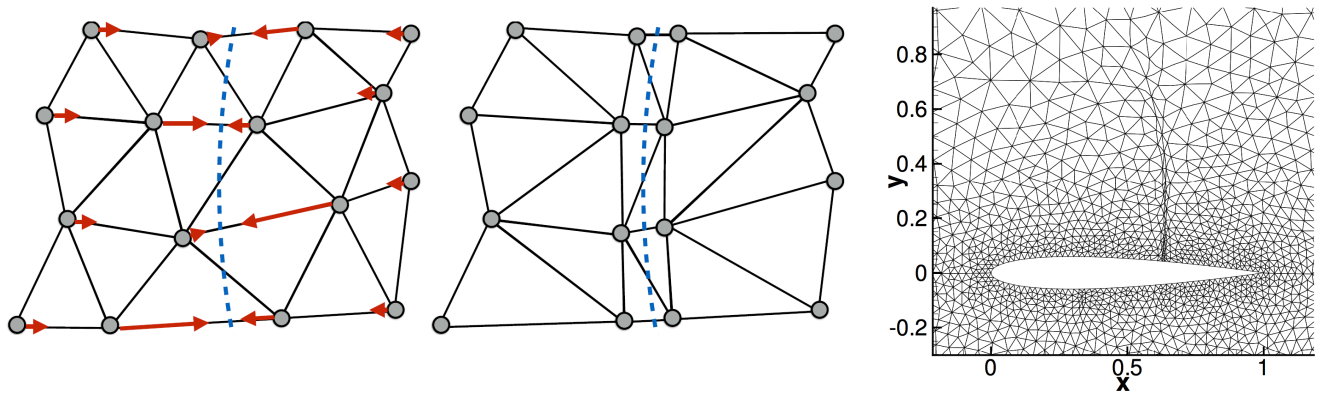


Figure 7.6: r-adaptation applied to resolve a shock [26]

adaptation [61, 110, 20] and initial a priori curvilinear mesh generation for complex geometries [106]. The main advantage of the method is keeping the mesh topology and the connectivity between the elements while creating a mesh matching the anisotropy of the flow features such as shocks, jets and boundary layers, see Figure 7.6. Moreover, by keeping the topology, one can devise a formulation, where no reassembly of the solver is necessary after the mesh movement which has computational benefits for the dynamic cases.

Complexity and Computational Cost However, this technique is usually the most computationally expensive one. Just considering the optimization algorithm, it requires minimizing an objective function, which typically involves assembling a linear system of all nodes. In an industrial test case, the GLL nodes reach $O(10^9)$, this prohibits the use of expensive algorithms. This becomes even more crucial when considering an unsteady adaptation. In Nektar++, Turner [106] used a variational formulation for minimizing deformational energy functional by a Gradient Descent algorithm, whereas MFEM opted out for minimizing the TMOP mesh-metric objective function through Newton’s method (MINRES) or L-BFGS gradient-based non-linear solver [20] with the latter being scaled to larger problems. Last, r-adaptation requires other control measures such as precise error indicator, min/max allowed deformation, integration with the CAD curve/surface for node-sliding, etc.

Limitations While mesh movement has been found to create anisotropic meshes matching the solution features precisely, the most significant constraint of the method is the fixed number of DoF. This limits the degree of refinement that can be achieved purely with r-adaptation. Depending on the complexity of the solution, this initial resolution may not be sufficient, as remarked by Marcon [61]. This restriction has been reported even for state-of-the-art r-adaptation mechanisms such as in MFEM [20] for ALE hydrodynamics and in Firedrake group Wallrock [110] for ocean modelling. Therefore, most of these authors usually recommend coupling the r- with h-refinement.

h-adaptation

The widely used h-adaptation was developed back at the end of the 80s and 90s with notable work by Lohner [59] for splitting and by Peraire [84] for remeshing. This approach can increase the resolution (refinement) when the larger elements are split into smaller ones or decrease the resolution (coarsening) when smaller elements are merged into a larger one. The method has been applied to both finite volume (FVM) and finite element methods (FEM) and has been the main workhorse for feature-based unsteady mesh adaptation due to its robustness, versatility and more importantly efficient parallelization techniques [59]. For example, the state-of-the-art implementation by Alauzet and Loisele from the INRIA group [57], [58], and the non-conformal work by Cervený in MFEM [14] or the most recent one by the DLR t8code [36] that is implemented in FLEXI. Despite all these beneficial properties that have been the main tool for adaptation around shocks, h-refinement has its own properties, which will be briefly discussed.

Concept of Conformity A conformal mesh is a mesh in which edges in 2D and faces in 3D have only two adjacent elements. In other words, it is a mesh without hanging nodes. This is a common problem for adapted meshes, where a hanging node is created on an arbitrary position of the edge, see Figure 7.7.

This mismatch is essential when calculating fluxes in DG and constructing mass and stiffness matrices for the shared faces in CG. Alternatively, one may avoid such a situation by using only conforming refinement operators, where not only the element of interest is split or collapsed, but also the adjacent one. If such non-conformal approaches are required, special treatment is needed at the interfaces for non-conforming elements in both CG and DG solvers.

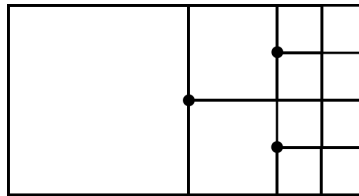


Figure 7.7: Non-conforming mesh with three hanging nodes.

In the DG SEM community, handling non-conforming elements for h-adaptation has gained traction in this decade, with solvers such as FLEXI, MFEM and deal.II, to name a few. As for the p-adaptation, the DG formulation is easier to account for the non-conformity on the shared face through the trace. The most common approach introduces a mortar space on the interface to weakly enforce flux consistency across the interface. This is achieved through L2 projection and integration over the non-conformal interface. Alternatively, one can use point-to-point interpolation, where the solution on the coarser interface is interpolated to create a conformal nodal space on the original interface for the flux evaluation. This is computationally cheaper

but has fewer numerical guarantees for mass conservation. For a more detailed comparison, the readers are referred to the work by Laughton and Moxey [51].

In CG SEM, this is a research area, to the authors' knowledge, with the only non-conformal h-adaptation for CGSEM being the Nek5000 version by Offermans and Peplinski [77]. In this work, they interpolate from the parent coarse element to the finer child ones on the collocation points but later disclose the need for a special multi-level Schwarz preconditioner. While successfully applied in the work by Massaro and Offermans [78], the core Nek5000 solver relies only on hexahedral elements, limiting its use to simpler geometries.

h-adaptation Operators The classical refinement and coarsening techniques for straight-sided meshes, developed mainly in the 1980s and 1990s, are summarised by Lohner [59]. These methods generally employ the following operators: edge splitting or removal, edge swapping, and edge insertion. For example, recursive subdivision of the largest edges is simple to implement by introducing a hanging node in the middle of the edge. Then, the element of interest is split 1:2 by edge insertion, and consequently, the adjacent elements are also split to preserve conformity. This approach can be iteratively applied to achieve a finer refinement. A similar approach can also be performed for coarsening. Last, the algorithm should ensure the conformity. Another alternative is an algorithm by Maubach that does not require additional swapping operator, but this has been found to be recursive and overrefined; hence, it has been limited to more academic use.

More recently, Loisel has developed a generalization of these mesh refinement operators under the name of cavity-based approach [58]. It first identifies a region for refinement through a continuous mesh metric field. Then efficient parallel node insertion in this cavity is performed, followed by local remeshing. The equivalent technique is used for derefinement. With the further developments for load balancing and partitioning, to the best of the knowledge of the author, this is currently the most widely used conformal anisotropic approach, capable of refining from 15 million elements to up to 1 billion on 120 CPUs for 25mins as of 2017 [57]. This approach has also been combined with level set methods for complete metric-based remeshing within the MMG tool [17].

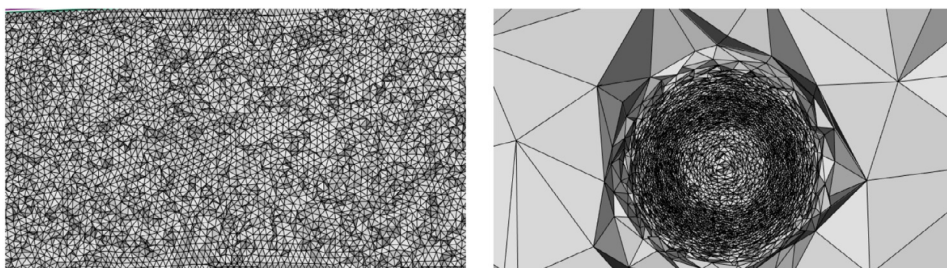


Figure 7.8: 3D blast simulation, uniform mesh (left), adapted anisotropic mesh (right). [57]

When high-order methods are of interest, there are a few worth discussing. First for non-conformal meshes we should mention the work by Cervený [14] for unstructured meshes and the work by Holke in the DLR t8code/p4est octree ones, both with a focus on performance with near exascale applications in MFEM and FLEXI respectively. In the end of this PhD project, Shi and Persson also presented a work on conformal tetrahedral curved h-adaptation [99]. When hybrid curved meshes are of interest, to the best of our knowledge, no open-source library is available.

7.6 Interpolation and Solution Transfer

The final step before restarting the solver is the transfer of the solution from the previous mesh at timestep $t-1$ to the adapted mesh. Generally speaking, one would favor a conservative approach that introduces minimal solution transfer error between adaptation cycles, as these errors will be propagated over the domain. This is of particular importance for the performance of unsteady problems with AMR, where using non-conservative approaches can lead to the need to run long "flush out" cycles to clean up the errors. The p-adaptation is the most straightforward, as the solution can be represented by hierarchical modal bases. Then for p-refinement, one just adds an extra expansion coefficient and zeros its value, which is conservative. Alternatively for p-coarsening, the unnecessary high-order modes are just dropped with minimal error from discarding highest order mode.

For r-adaptation, where the topology is also preserved, one can either apply a global interpolation or Arbitrary Lagrangian-Eulerian(ALE) approaches. The interpolation approaches are easier to implement, but generally non-conservative and sensitive to solution smoothness. Therefore, particularly in DG solvers, ALE approaches are preferred where the mesh grid velocity is estimated and the conservation is achieved through flux corrections [20]. Last, for h-adaptation, one can directly use local interpolation in the case of refinement, which can be implemented conservatively. However, when coarsening, swapping or smoothing are added, non-local, non-conservative interpolation operators are employed.

Finally, one can point out global conservative approaches such as the supermesh by Farrell [29], where both accurate and conservative solution transfer is achieved yet computationally expensive. As a proof of concept in this work, the hierarchical modal basis approach will be employed for the p-adaptation and standard non-conservative global interpolation is used for the h and r-adaptation.

7.7 The need for hp-adaptation

While high-order methods have been in development for decades and are currently on the brink of industrialization, they are often advertised with their spectral convergence, lower dissipation, etc. However, for someone interested in complex geometries, probably the most beneficial feature is the capability to use all three adaptation techniques for creating a true hp/SEM method. In the past 10 years, the community is slowly realizing this powerful feature and working towards it. To understand some of our decisions later in this part, we will discuss the applicability of the different techniques for the smooth and discontinuous flows in an ideal full hp/SEM framework, taking advantage of the vast literature in the field.

7.7.1 P-adaptation for smooth flow regions

One of the most important benefits of using high-order methods, particularly the spectral/hp element method, is its spectral convergence property. This translates to the truncation error ϵ_{tr} being sensitive to the smoothness of the solution u , the size of the element h and most notably the polynomial order p . In the special case, where a smooth solution is present, this implies an exponential convergence of ϵ_{tr} . Even though the computational cost might be higher than the theoretical one, the property clearly shows the advantage of mesh refinement by increasing the polynomial order p in comparison to adding DoF by decreasing h and increasing the number of elements [43]. Adding on top of all these qualities state-of-the-art p-adaptation, hence anisotropic variable p , Kompenhans & HORSE3D achieved an overall decrease of up to 80% of number of DoF for the inviscid Euler ($M=0.3$ $AoA = 0^\circ$) simulation of NACA 0012 airfoil in comparison to a uniform $p=8$. [49] This group has since extended the work also for larger 3D dynamic p-adaptive cases, driven by NN indicators. Moreover, the community has consistently shown that the SEM solvers achieve the highest throughput at very high polynomial orders, ex $p=6-7$, as discussed by Moxey[69], meaning such p-adaptive schemes applied on the coarse elements downstream might be even more computationally efficient.

$$\epsilon_{tr} \simeq h^p \left(\frac{d^p u_h}{dx^p} \right) \quad (7.5)$$

The drawback of the higher polynomial order is that they generate high-frequency numerical oscillations in advection-dominated flows with (non-smooth) discontinuities, where dispersion errors are transported through the domain, grow in time and could cause the simulation to diverge. This is also known as the Gibbs phenomenon and needs special treatment by adding either explicitly or implicitly diffusion in the vicinity of these regions or filtering the high-order terms, which hinders the previously discussed benefits [86]. Examples of such flows with discontinuities are shocks and contact discontinuities in compressible flows and shear/mixing layers in jet flows. Hence, it is not feasible to rely completely on p-type refinement over all

flow features. One option is to combine a feature-based smoothness error indicators to identify these zones and then maintain a low order p there. This has been investigated, for example by Ekelschot on a NACA 0012 airfoil [27]. However, the necessity for combining p with r or h refinement in these regions was concluded.

7.7.2 h and r -adaptation for discontinuous flow features

Naturally, one can decrease the polynomial order of the simulation in the regions of the discontinuities and combine it with r -adaptation in this region. Marcon has proposed this with a smoothness (shock-sensor) indicator for capturing the shocks and reported improved performance for NACA0012, but heavily relying on the initial mesh resolution being already fine enough for the rest of the flow [61]. When extended to an unknown case, this approach heavily underperformed and was not able to resolve the complex shock-shock interactions due to the lack of DoF to cluster around the shock.

Hence, the majority of the hp -adaptation techniques heavily rely on h -adaptation to provide the additional degrees of freedom to capture the discontinuities when starting from a coarse mesh. For example, the work by Basile for hp -adaptation, where the h -adaptation was performed by remeshing with MMG and then p -adaptation internally [6] again with a smoothness indicator for distinguishing between the features in a quasi-unsteady manner.

The most advanced hr -adaptation, however, is the one by the MFEM group, see the work by Dobrev et al. [22], which uses the TMOP and Hessian indicators to drive a multi-material hyperbolic problems dynamically. On top of the complexity of the ALE multi-material hydrodynamics, they first introduce additional DoF into a very coarse mesh and then move the mesh with high-order r -adaptation. This mesh movement is executed iteratively on the GPU device, and only once the mesh is too distorted or tangled, then a new remeshing/ h -adaptation cycle is performed on the host CPU.

7.7.3 State-of-the-art

Over the course of this PhD, the h , hr , and hp -adaptations have gained traction, and some powerful AMR tools have been developed in the open-source community. Therefore, a brief summary of some of the state-of-the-art AMR and high-order solvers is provided in Table 7.1 with their presented capabilities.

Table 7.1: Capabilities of AMR state-of-the-art simulation frameworks

	h	NonConf	r	p	hr	hp	hrp	Error Estimator	Anisotr	Driver	Solver
Nektar++	-	-	+	+	-	-	-	Smoothness/Adjoint	-	Naive	PIP Interpolation
INRIA	+	-	+	-	+	-	-	L_P -norm/Adjoint	+	Metric	Wolf FVM
MFEM	+	+	+	?	+	-	-	Variable	+	TMOP	DG CEED/BLAST
HORSE3D	-	+	-	+	-	-	-	Truncation	+	CNN	DGSEM Mortar Hex
FLEXI	+	+	-	+	-	-	-	Truncation	+		DGSEM Mortar Hex

7.8 Research Objectives

The properties of the Spectral Element Methods do provide some clues on what could be more efficient. However, to this date, no full integration between all mesh modification techniques has been performed in any of the major SEM codes. Hence, this part of the PhD project will try to provide a relatively low TRL proof-of-concept on combining all mesh modifications with the mind of complex geometries and their future combination into a single automatic framework for *a posteriori* mesh adaptation. While this will be by no means a complete *a posteriori* AMR framework, we aim at providing the mesh modification capabilities and showcasing the potential of this integration for both CG and DG solvers in both compressible and incompressible flows. Based on that, we can establish the following research goals:

1. **Provide High-Order self-contained and scalable h-adaptation for curved 2D and 3D meshes.**
2. **Investigate the adaptation capabilities on very coarse, underresolved meshes outside the asymptotic region and their capability to improve the solution accuracy resolving the correct flow physics.**
3. **Investigate if the combination of all mesh modification techniques into a single h-r-p framework is indeed a more efficient choice than standard h, hp or hr approaches.**

Chapter 8

High-Order Curvilinear h-adaptation

This chapter proposes an isoparametric approach with three local mesh modification operators: high-order splitting, high-order swapping and smoothing. The h-adaptation adopted in this work is conformal and is aimed at triangular or hybrid triangular-quadrilateral meshes in 2D and hybrid prism-tetrahedral ones in 3D. The methodology starts with a very coarse boundary-conforming curvilinear mesh as in Part I and the adaptation can be performed either on the initial base mesh or the most recent refined mesh. The h-adaptation approach implemented in this work has four goals. First, achieving a maximum degree of locality on the operators to ensure easy future parallelization. Second, we aim to perform all mesh modification techniques directly on the curved mesh, avoiding the slow and costly CAD-API projection but simultaneously preserving the geometrical accuracy and quality of the mesh. Third, the methodology should be generic from a solver perspective and should apply to both conformal and non-conformal solvers. Last, while not the focus of this thesis, the process should potentially allow for seamless coarsening as is required for dynamic AMR.

The combination of these goals and the mesh generation intricacies from part I, means that to the best of our knowledge, the isoparametric high-order splitting, augmented by some mesh quality improvement operators, is the only approach capable of satisfying all these in the highly distorted curved scenario. While the cavity approaches pose an interesting alternative, at the moment of starting this project, the lack of element generality and the use of straight-sided fully tetrahedral meshes deemed it unacceptably expensive for maximizing the computational efficiency of high-order methods in the long term with the variable prismatic boundary layer meshes.

8.1 High-Order Isoparametric Splitting

The core of the h-adaptation is an efficient isoparametric splitting operator. To achieve the maximal level of locality, the splitting is performed hierarchically - first the edges, then the faces and finally the elements. This way, one can track directly the elements affected by a lower-level split and then ensure final conformity. Irrespective of the hierarchy level, when a curved mesh object is present, the operation is performed in the reference element space(ξ) based on a pre-defined standard map. The child-adapted elements remain curved on their own, inheriting many of the properties of their parent element.

Algorithm 3 High-Level Mesh Adaptation Algorithm

- 1: Identify Edges/elements/regions to split
 - 2: (Mark Elements for adaptation) (If Conformal)
 - 3: Split edges
 - 4: Split faces
 - 5: Split elements
-

Assuming that the initial coarse mesh is valid ($J_{sc} > 0$) and with sufficient geometrical accuracy, see Part I, the isoparametric splitting approach provides several benefits:

- *Flexibility.* This splitting approach allows one to drive the splitting by many different criteria - the element IDs, the longest edge of an element, uniformly, by target area field or by the finest map. The non-conformal adaptation is inherently achieved by just having a single reference splitting map per type of element (prism, hex, tet, pyramid) and adding a new type is just implementing a new map that describes how the edge splitting transfers to the construction of the new adapted child elements. This is not straightforward for the conformal case, but it still comes to implementing new maps while keeping the overall module, performance and driver exactly the same.
- *Mesh Quality.* The main benefit of the proposed splitting is the theoretical guarantee of validity for the resulting elements. This follows from the splitting inside the standard element Ω^{st} . Moreover, if the finest map is used, where all elemental edges are split, the child triangles will have similar or better quality in terms of scaled Jacobian than their parent. This automatically translates to bypassing the need for untangling curved elements and hence not applying costly global approaches such as the Variational Optimizer from Part I.
- *Geometrical Accuracy.* The decisive factor for selecting h-adaptation was the preservation of the element curvature due to the splitting in the reference element and then performing a backwards transform to the nodal physical space of all edge and face curvature from the parent to the child adapted elements. This way if the initial element is valid and with

good geometrical accuracy, one can preserve the geometrical accuracy directly, without any additional treatments.

- *No need for CAD-API.* The combination of keeping the geometrical accuracy and the mesh quality allows us to bypass any CAD operations during the adaptation process without sacrificing accuracy. As explained in Part I, these are usually expensive Newton optimization problems performed by third-party software, requiring full CAD processing, which, on top of being suboptimal for performance, could tangle the mesh.
- *Parallelism and Scalability.* Last, while out of the current scope, the hierarchical approach used in the implementation allows for seamless parallelization, both in a shared memory (OpenMP) and segregated manner (MPI) inside the partition itself. Such an approach has been shown to scale and perform well in a non-conformal manner for fully hexahedral or full tetrahedral meshes by Holke [36]. In our generic implementation, where conformity propagation is activated, the only communication happens about the shared edges in 2D or faces in 3D for the element marking. Nektar++ already possesses such MPI communication infrastructure for the DG traces, hence a future parallelization (not part of this work) is accounted for.

8.1.1 h -adaptation driver

In both 2D and 3D, our implementation allows for selecting the elements to be split in multiple different ways: For example, one can split all elements uniformly, pass a list of Element IDs, Edges IDs, Target Area Size or even the complete Nektar++ solution field with a threshold to be adapted. One can choose the splitting for these elements to be driven by the longest edge, by maintaining uniformity through area grading, or by splitting all element edges, which corresponds to our finest case. Here, we always use the finest map of the initially selected element for the triangular and tetrahedral elements. Last, one can select if the conformity should be propagated in the mesh or not and if a minimum edge length or face area/volume is enforced.

8.1.2 2D Isoparametric Splitting

The 2D process starts with selecting the elements to be adapted. Then, their edges are split in the high-order isoparametric fashion. In 2D, this corresponds to an isotropic 1:4 triangular map, 1:4 isotropic quadrilateral and two anisotropic 1:2 maps. The whole 2D splitting process for a curved triangular element is visualized in Figure 8.1. At this point, if one is interested in conformal splitting, the adjacent elements to the initially selected ones are added to the

list to be split. As they will have different combinations of edges to be split, additional maps are required. In 2D, this corresponds to overall 10 maps for triangles (1:2,1:3,1:4) and the 3 already described ones for the quadrilateral elements (1:2, 1:4). Then, by looping over the selected elements and identifying the desired map, we can adapt them in a completely local fashion. Please note that the internal edges in the 2D case are created in the reference element of the macro element, making them curved as well, which is what brings us the theoretical guarantee for mesh validity [74]. For efficiency reasons, an additional check is performed for every split parent element: If it is fully straight-sided, the splitting operator avoids the costly transformation and constructs the child vertices directly in physical space through an affine interpolation between the parent vertices.

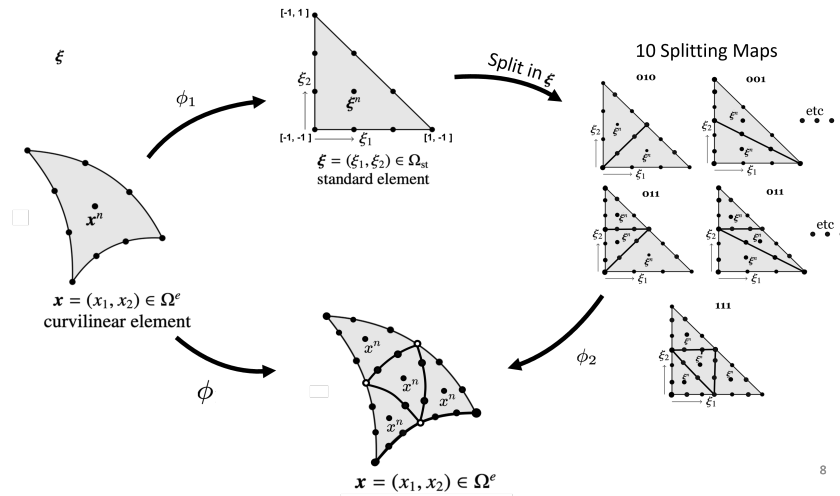


Figure 8.1: Mechanism behind the isoparametric splitting, from the coarse curved element (1:4 reference map) to four curved and valid child elements. First, the curved parent element transformed to standard space ξ_1, ξ_2 using map ϕ_1 , then a predefined map is selected based on which edges are split. This gives the vertices, edges and high-order nodes of the child elements in standard space of the parent element. Finally, one can map the nodes location in physical space and construct the child mesh objects using the ϕ^{-1} also in physical space.

8.1.3 3D Isoparametric Splitting

Our philosophy in 3D is similar. First, we use one of the drivers to select the edges, elements, etc., that are marked to be split. Again, we will describe it with the finest map concept, where all edges of the initially selected elements are marked for splitting. For the hybrid tet-prismatic meshes, this corresponds to a 1:8 map for the tetrahedral elements and 1:4 for the prismatic ones to propagate the tetrahedral splitting. The $y+$ control will be left out of the discussion for simplicity, as some of the concepts were already described in the mesh generation part of this thesis.

With these two maps, one can already achieve a uniform refinement over the whole domain but also non-conformal curved refinement, similar to what Holke proposed in t8code at the

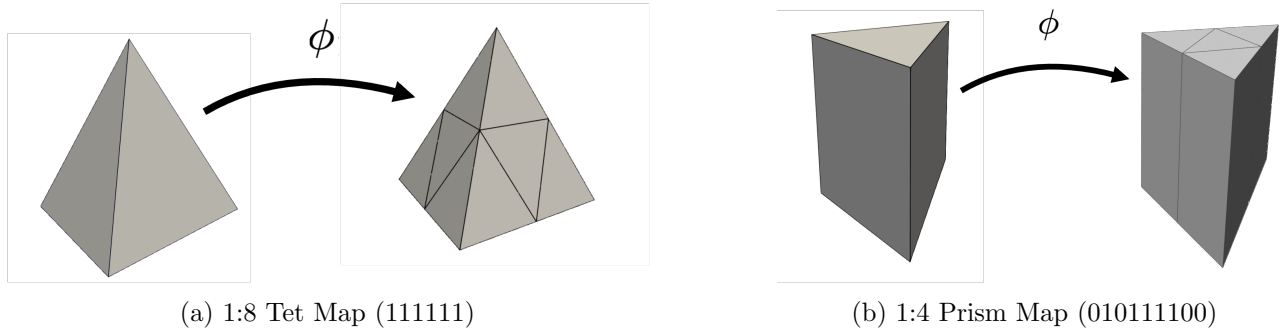


Figure 8.2: High-Order uniform refinement (1:8 tetrahedral and 1:4 prismatic reference maps) showing transformation from the coarse parent element to the split child elements.

end of this PhD project. However, our goal is also to target CG applications with conformal refinement.

Conformity Propagation The process in 3D can be derived in a similar way as for 2D: First, identifying the edges to be split and, from there, marking all adjacent to this edge element for adaptation. Then, by splitting these edges isoparametrically and identifying the map of the tetrahedron, one can split the list of faces and, ultimately, split the elements. However, when a triangular face has two split edges, this creates two alternative reference configurations, which becomes important in 3D and complicates the propagation of conformality. Therefore, such a brute force approach as in 2D having all possible alternative maps in 3D is not straightforward as all possible combinations end up being around 240. In the FVM literature, it has been demonstrated that these can be reduced through the use of similarities that employ symmetry. However, this does not account for the fact that in SEM, as of Nektar++, the collapsed coordinate system in CG mandates keeping the orientation of the triangles to be maintained after the split to ensure interface consistency. [43] Therefore, the orientation of the elements need to be preserved after the split with appropriate reference maps or the reordering of vertices.

Nevertheless, we adopt a conformity propagation algorithm that makes it possible to avoid implementing all possible maps. This comes at the cost of splitting a minimal number of additional edges that ensure known maps for all elements, and it works as follows: The initial high-order split of the selected edges is performed according to the user-defined driver. Afterwards, all adjacent elements to those edges are shortlisted through a hashmap, and the reference map of those tetrahedra is identified.

Suppose any of the desired maps is unavailable for a particular adjacent element. In that case, the algorithm either splits all edges or only one more edge, which naturally extends the list of elements to be split, hence requiring checking an extended list iteratively until all elements have a combination of edge splits that is present in the reference map list, see Algorithm 4. From this moment onwards in the module, all operations are local and are guaranteed to have a valid, already implemented map - conformity is ensured.

Algorithm 4 Conformity Propagation Algorithm**Require:** Initial Edges to be split E_{init} , user-defined driver**Ensure:** All Elements Have Edge Split Configuration present in the Reference Maps

```

1: Hash Map EdgeToElements
2: Edge Split  $E_{\text{init}}$ 
3: while New Edges To Split do
4:   for all  $E_L \in \mathcal{T}$  do
5:     Identify  $E_L$  Map
6:     if Map unknown then
7:       Add "NewEdgesToSplit".
8:     end if
9:   end for
10:  Split NewEdgesToSplit
11:  Add to ElementsToAdapt
12: end while

```

The initial implementation and the first subset of tetrahedral and prismatic maps necessary for conformal refinement were implemented by the author. These were sufficient to ensure conformity after multiple iterations of propagation. However, often, the propagation was excessive and expensive. Since then, together with Jingtian Zhou, we have extended this to over 130 tetrahedral maps, thereby drastically reducing the amount of refinement propagation. At this point, our implementation captures all prismatic maps for conformity propagation, all tetrahedral ones with 1,2,5,6 edges split and the majority of 3-edge maps. Hence, in our practice, the remaining maps with four edges split and just adding one extra edge for splitting is sufficient to bring the number of IdentifyEdgesToSplit iterations up to a maximum of 6, with the majority of cases having between 0-2 iterations.

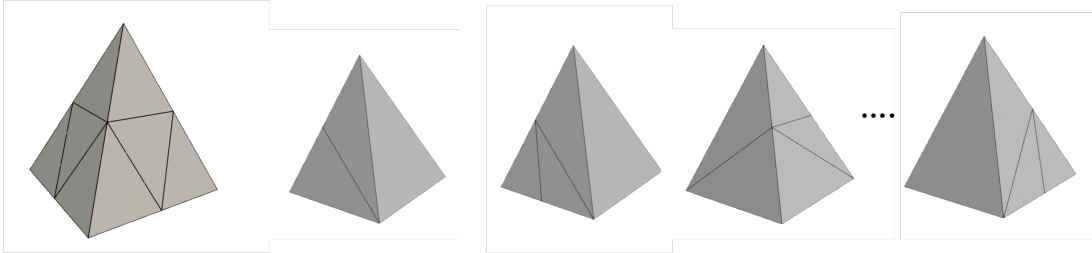


Figure 8.3: Some of the 130 alternative maps implemented in our high-order h-adaptation. Please note, symmetries or alternatives were used in the hope of avoiding reordering tetrahedra for the Nektar++ collapsed coordinate system.

8.2 Quality Improving Operators

We presented some of the advantages of this approach, especially in the high-order context. However, it is also criticised for the deterioration of mesh quality when used for multi-level adaptation. Therefore, usually linear straight-sided swapping and smoothing are applied. In this section, we describe our approaches for high-order swapping in 2D and smoothing in 2D and 3D.

8.2.1 High-Order Swapping

The second operator, part of the h-adaptation, is edge swapping, which aims at improving the overall mesh quality after the elemental splitting. Opposite to the classical mesh adaptation techniques, where the h-adaptation is performed on the straight-sided meshes, we aim to extend it using the swap on the curved elements, both on the boundary and internally within the domain. Hence, we need an operator that can propagate the benefits of the isoparametric splitting without sacrificing mesh validity and geometrical accuracy - isoparametric swap.

The process looks as follows - first, the elements affected by the previous adaptation are marked. In 2D, the longest edge of the element Ω^e is identified, together with its adjacent element Ω^A , forming an elemental pair, see Figure 8.4. This pair represents a macro quadrilateral Ω^Q , with the edge being the current diagonal D_0 . Before forming this quadrilateral, we check if the swap in the diagonal will improve the quality of the mesh through the vertex valence and the elemental angles, and if Ω^Q valid, we create its isoparametric representation. At this point, the diagonal is swapped in the standard/reference element Ω^{st} . Finally, the algorithm creates two new triangles with the same external edges but with the new curved diagonal, which improves the overall quality of the mesh as seen in Figure 8.4 b).

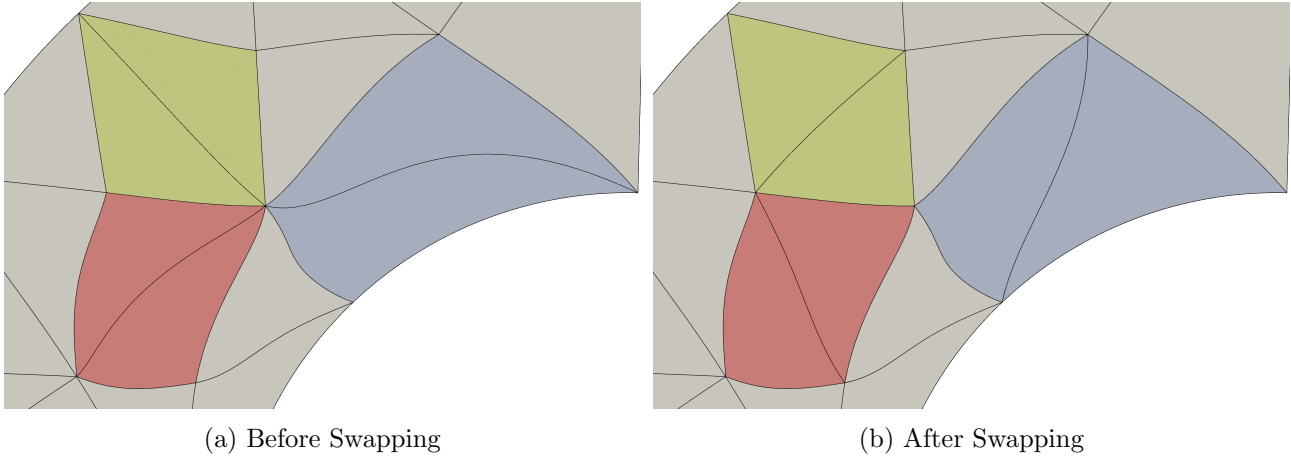


Figure 8.4: Highly deformed region after isoparametric splitting with initial vertex valence of 8 and the same region after isoparametric swapping in three element pairs resulting in vertex valence of 5 and improved mesh quality.

Given a valid quadrilateral candidate Ω^Q , this approach provides a theoretical guarantee of the validity of the final mesh after the h-adaptation split-swap workflow. Similarly, the geometrical accuracy of the boundary elements is also preserved as the external quadrilateral edges are reused directly. Please note that often, swap operations lead to elements with high deformation like their parent elements, which is not strictly necessary for the overall mesh quality after the swap, see the top blue element on Figure 8.4 b). Hence, if beneficial for the mesh quality, some of these edges can be linearised (transformed to straight-sided ones) in a final sweep over the affected elements. While it is possible to be extended to 3D, the 3D swapping operator has not been implemented at this stage.

8.2.2 Mesh Smoothing

By construction, these splitting or bisection adaptations assume starting with a relatively uniform coarse mesh and the child elements will always attain the topological aspect ratio and skewness of their parent. Consequently, in the case of complex geometries, non-uniform initial mesh or even just after multiple adaptation cycles, they tend to become distorted and have a steep volume transition between regions, see Figure 8.5. This is usually addressed through a smoothing operator.

In the high-order sense, we can use our Variational Optimizer and its extension to r-adaptation for this purpose in both 2D and 3D. However, we also implement a standard Laplacian or centroidal smoothing operator that can be applied as in a linear element. While not as robust and not sliding the boundary nodes, this approach is substantially cheaper and as we will see later, sufficient for many of our current applications.

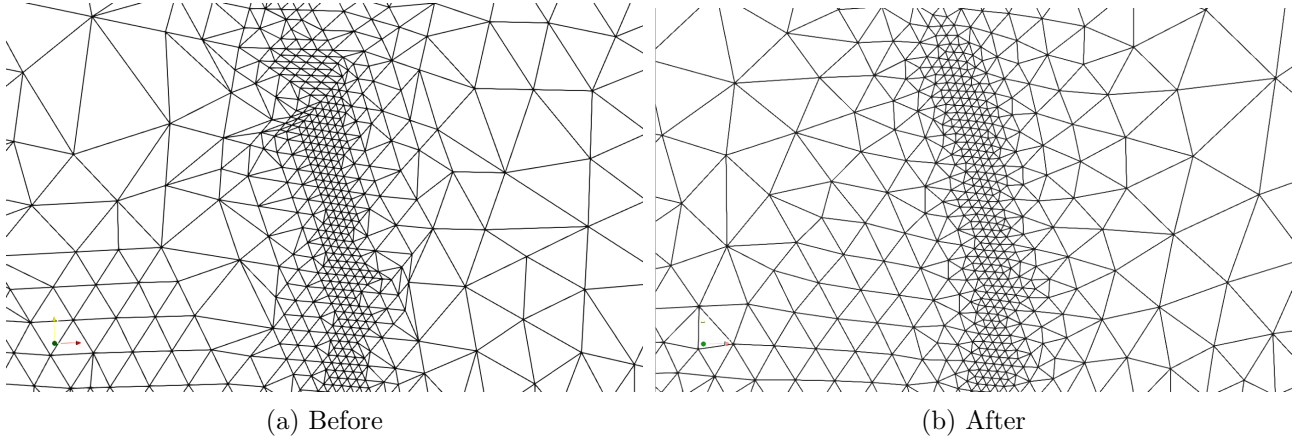
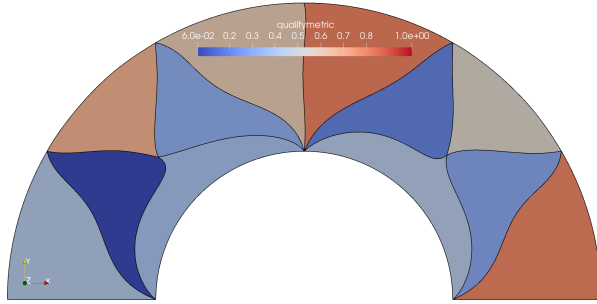


Figure 8.5: The effect of mesh smoothing in a region after mesh swapping resulting in an improved overall volume growth rate between elements.

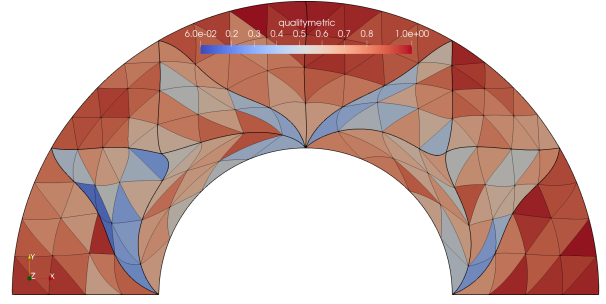
8.3 2D - Verification & Validation

8.3.1 Uniform Adaptation, Geometrical Accuracy, Mesh Quality and Performance

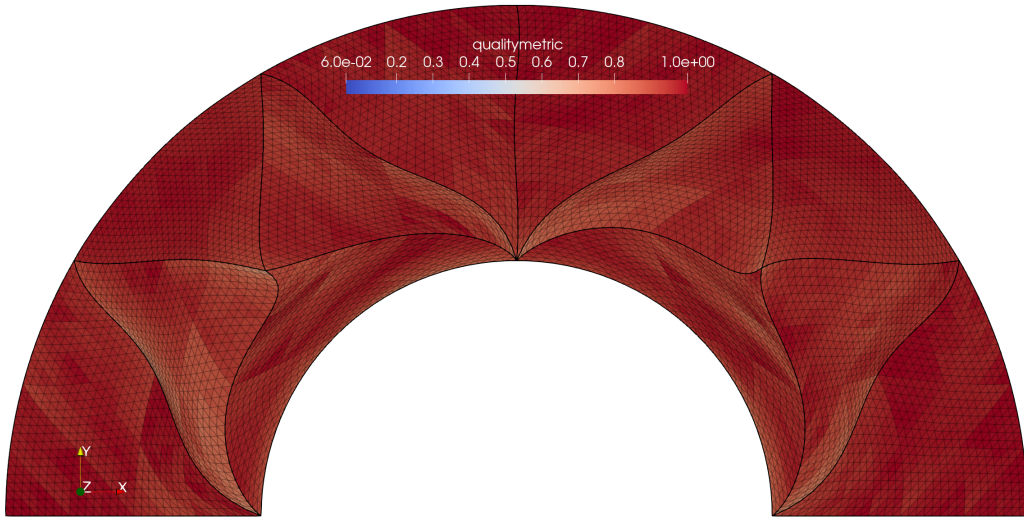
To validate some of the previously described benefits, we select a simple ring with $R_{in} = 0.5$ and $R_{outer} = 1$. We generate an initial highly distorted curved mesh with $p=4$ and only 12 elements with NekMesh from a CAD. This was intentional and will prove helpful in analyzing the geometrical accuracy, quality and even performance. Afterwards, the uniform h -adaptation has been used multiple times.



(a) Initial



(b) hh



(c) hhhh

Figure 8.6: Uniform Isoparametric Refinement of a highly distorted, but valid, initial $p=4$ mesh. Please note the improvement of the minimum J_{sc} from 0.06 to 0.17 and finally to 0.67 for the hhhh-adapted mesh.

Mesh Quality For this test, the initial highly distorted but valid mesh with minimum $J_{sc} = 0.06$ provides an opportunity to analyze the mesh quality when the mesh is refined uniformly, see Figure 8.6. As expected, after the refinements the child elements are curved, and the mesh is kept valid, which is not the case if the splitting is performed in a straight-sided fashion. More consequential is the fact that the overall mesh quality of the mesh improves monotonically over the refinement cycles. Looking for the most distorted element even after only two refinement steps, we observe improvement on all elements and minimum $J_{sc} = 0.17$. If we continue further, this reaches minimum $J_{sc} = 0.67$ for the 5-level refinement. We hypothesize that this comes from the change of the maximum J_i , internal to the element, rather than an increase in the minimum internal J_i , as the maximum deformation will still be caused around the sharpest angle. Despite that, this demonstrates that using the finest possible map for the elements of interest will not only increase the resolution but also the quality of the mesh for those elements. Consequently, it is adopted throughout this thesis. However, this quality increase is not strictly present for the conformal propagation maps, hence the use of swapping and smoothing.

Geometrical Accuracy The extremely coarse mesh with only two elements on the R_{in} allow us to assess the preservation of the geometrical accuracy not only with the CAD-reconstruction but also analytically. Initially, $\delta_{max} = 4.5 \times 10^{-5}$. After 5 cycles of adaptation, the error has stayed $\delta_{max} = 4.5 \times 10^{-5}$ with a difference of $O(10^{-8})$, which is three orders of magnitude lower than the initial δ_{max} , hence for any practical purposes, it can be concluded that the geometrical accuracy is preserved.

h-adaptation performance As an initial implementation our h-refinement is currently single-threaded, but the methodology and the implementation were designed with a maximum degree of locality in preparation for the planned parallelisation of NekMesh within Nektar++. Moreover, the refinement is performed according to the maximum order inside each individual element and not the highest order of the mesh. This ensures good performance while allowing some very high-polynomial mesh elements to be present either due to the geometry or the flow feature in an ALE setting, for example. To assess the performance, we use the same test cases, where all elements are deformed with a uniform polynomial order and are marked with the finest 1:4 map. This can be seen in Figure 8.7. Refining a p=1 straight-sided mesh is the cheapest due to the lack of creating the expansion and the forward and backwards mapping of the isoparametric element. Consequently, in the curved case, the cost increases with the increase of the basis function representing the geometry. What is important to note is that in 2D, the splitting operator scales linearly and even in serial it requires $t < 1s$ for 200×10^3 elements. Moreover, the difference between the straight-sided split and a high-order p=5 one is only 2.5 times, which encourages the use of h-adaptation in comparison to costly mesh movement r-adaptation or remeshing.

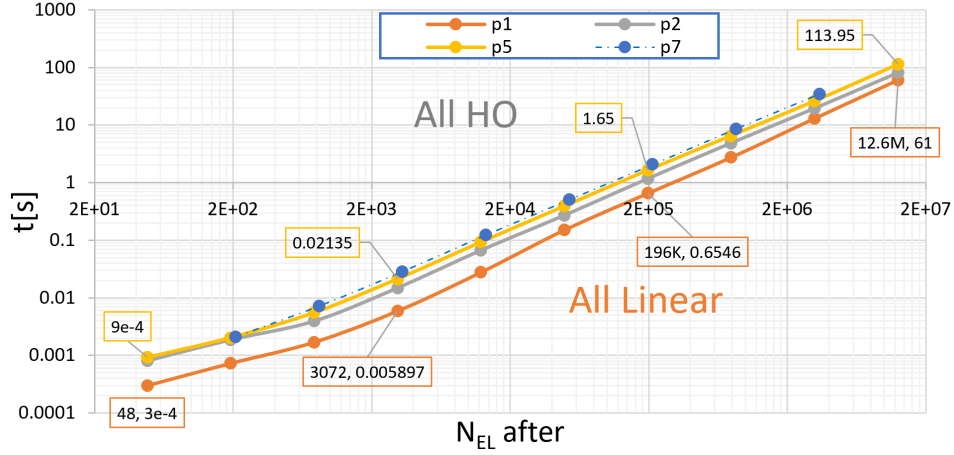


Figure 8.7: Worst case serial performance for uniform h -adaptation for different polynomial orders. Please note the timings do not account for the I/O time of NekMesh.

8.3.2 Conformal h -adaptation

To showcase the conformal propagation in 2D, the same ring geometry is used. In this case, we prescribe an infinitely thin region that should be refined and is advected in the domain with $R \in [0.5-1]$ emulating the wavefront. In this case, we start from a more refined case, see Figure 8.8 a) and assume that four levels of refinement are sufficient for the test. The split-swap-smooth process is applied, and the results can be seen in Figure 8.8. From this adaptation, one can observe the use of the other alternative maps as well that ensure the conformity of the mesh. This was validated both visually and with the internal NekMesh tools, where it is ensured that every internal edge shares exactly two elements. Achieving the same four-level refinement with a uniform mesh would require between 15 to 21 times more elements at the different stages of a potential simulation.

This process consisted of four level split-swap-smooth calls, which are currently performed offline and hence, the overall performance declines with an increase of R , reaching around two seconds from base to final mesh. Such iteration includes 12 I/O calls, accounting for $> 90\%$ of the cost. Hence, a recommendation of this thesis is the efficient integration of NekMesh within the Nektar++ data structure to avoid the I/O and transition to MPI-based parallelism.

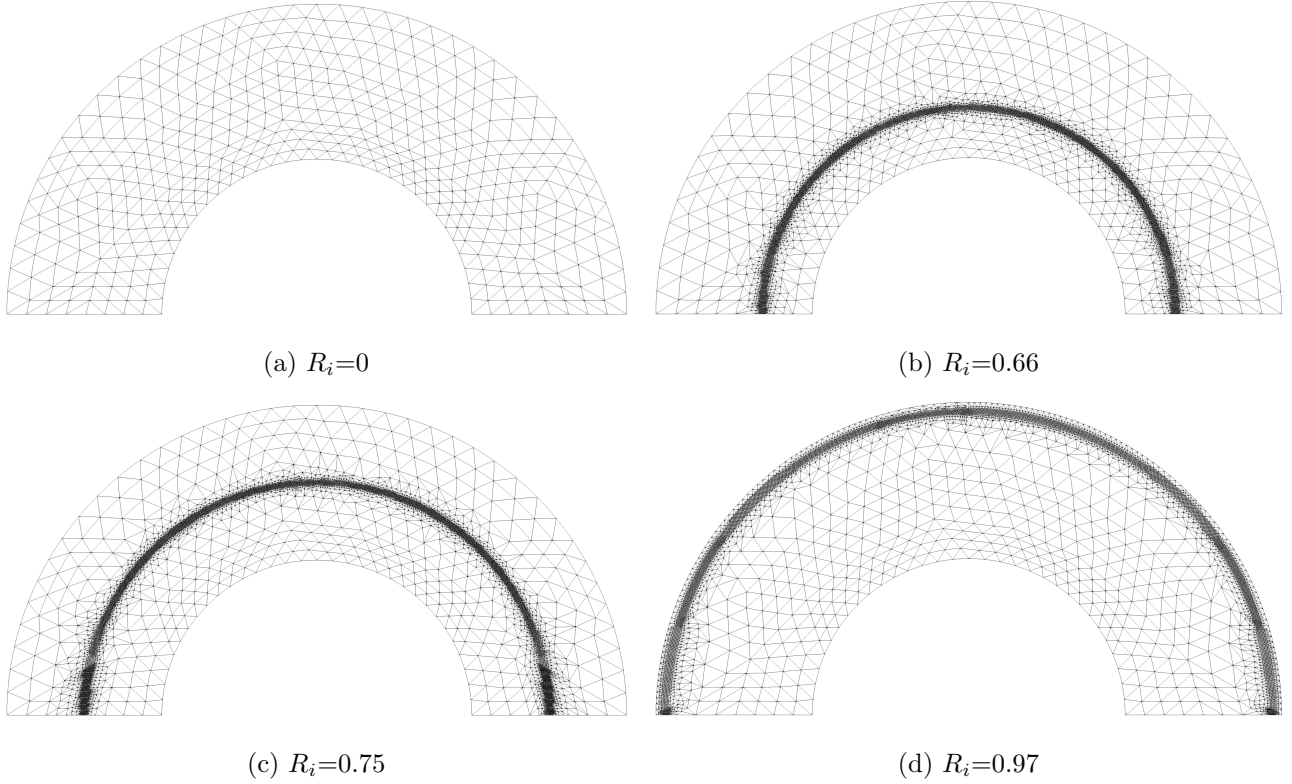


Figure 8.8: Conformal h-adaptation targeting a thin advected front in the semicircle geometry. The process consists of four level of split-swap-smooth calls on the base mesh (a) for every one of the advection stages (b),(c) and (d). Please note the conformity is preserved and mesh valence has not drastically deteriorated in every stage of the process.

8.4 3D - Verification & Validation

8.4.1 Uniform Adaptation, Geometrical Accuracy, Mesh Quality and Performance

In 3D, a very coarse high-order deformed academic geometry is used for initial validation. In this case, a $p=4$ semisphere with $R = 0.25$ and minimum $J_{sc} = 0.24$ from 4452 tetrahedral elements with 4% of the elements being deformed. We refine the mesh uniformly with the finest map, in this case 1:8, where all edges are split. Similar to the 2D case, the mesh validity is preserved through all three refinement cycles. After the adaptation, the majority of the elements do improve their mesh quality significantly with a minimum $J_{sc} = 0.29$. While a lower improvement compared to 2D, in terms of minimum J_{sc} , this is still expected to contribute to better solver performance.

More significant for the solver performance, the algorithm does distinguish curved and straight-sided elements in the split, keeping the relative number of curved elements the same of 4%. This leads to a strong scaling of the splitting near linearly - (0.45s (3.5×10^3) - 3.6s (2.85×10^5) - 32s (2.3×10^6)). In this we do not include the I/O time, but do include some core NekMesh modules that do not scale linearly and deteriorate the performance for larger meshes. Moreover,

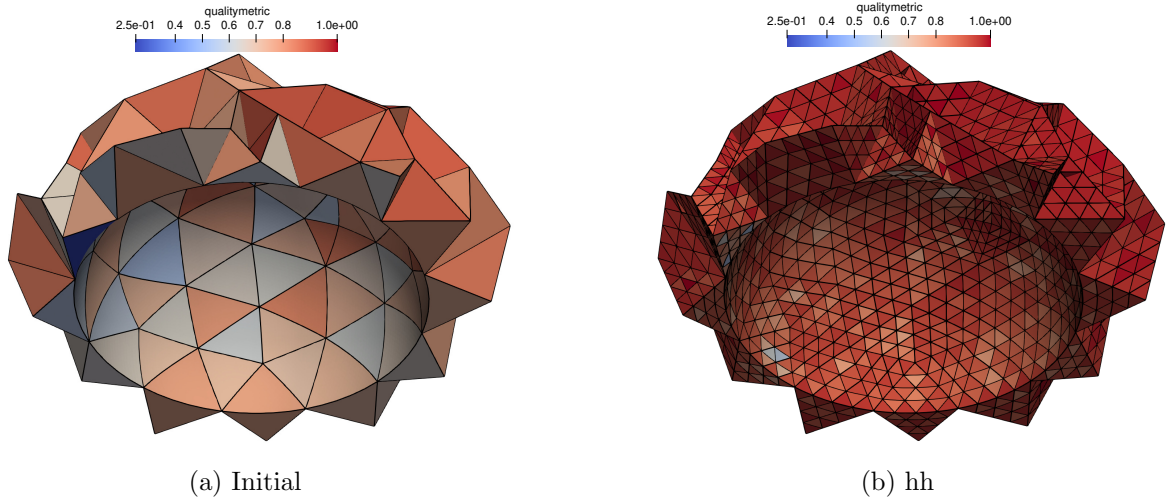


Figure 8.9: Uniform Isoparametric Refinement of a semisphere mesh. Please note only the curved elements of the mesh are presented for clarity purposes.

when hybrid meshes with tetrahedra, prisms and pyramids are used we do need to apply an additional reordering which is Nektar++ specific. Therefore, while acceptable for the tested meshes with refinement cost ($> 5\Delta t$), this clearly shows the need for using the Nektar++ MPI infrastructure for parallelization and optimization of core NekMesh utilities if dynamic refinement is of interest.

Finally, the curvature of the surface mesh is maintained throughout the meshes with a minor deterioration of $O(1 \times 10^{-5})$. This supports the hypothesis from part I of the thesis that if a very coarse, accurate high-order mesh can be produced, then no CAD interactions are necessary afterwards and the desired x^+ and y^+ can be controlled inside NekMesh through mesh modifications for complex geometries. A second application showcasing this exact application can be seen in Figure 8.10, where the eIFW tetrahedral mesh is split isotropically from a very coarse one through two-level h -refinement. No CAD operations were used in this example, showcasing the applicability to complex geometries with non-uniform curvature.

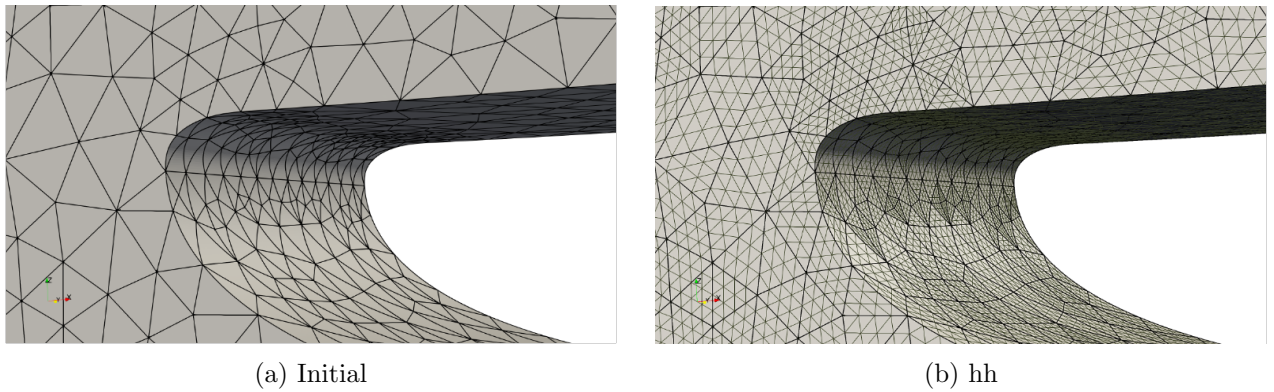


Figure 8.10: Uniform Isoparametric Refinement of the eIFW tetrahedral mesh. No CAD-operations were used and the curvature of the base mesh was propagated through the high-order refinements.

8.4.2 Conformal h-adaptation

To demonstrate the conformity propagation, another case is created similar to the one in 2D, but this time a cube. Here, we start from a finer mesh with 24576 elements and advect the adaptation zone as for the 2D one, increasing the radius R . In 3D, we use the h-adaptation through three-level splitting. The resulting meshes can be seen in Figure 8.11. Over all "timesteps" the mesh is conformal and well capturing the moving front with $\Delta x = h/8$. One can also observe the artefact of the high valence vertices due to the conformity propagation maps. While this effect is exaggerated due to the infinitesimally thin refinement region, this is an undesired and known limitation of the method. Therefore, the author recommends an extension to include high-order swapping also in 3D, constraining the refinement up to three levels of refinement in a thin region - for example, in shocks or interface tracking and the use of implicit time-stepping techniques.

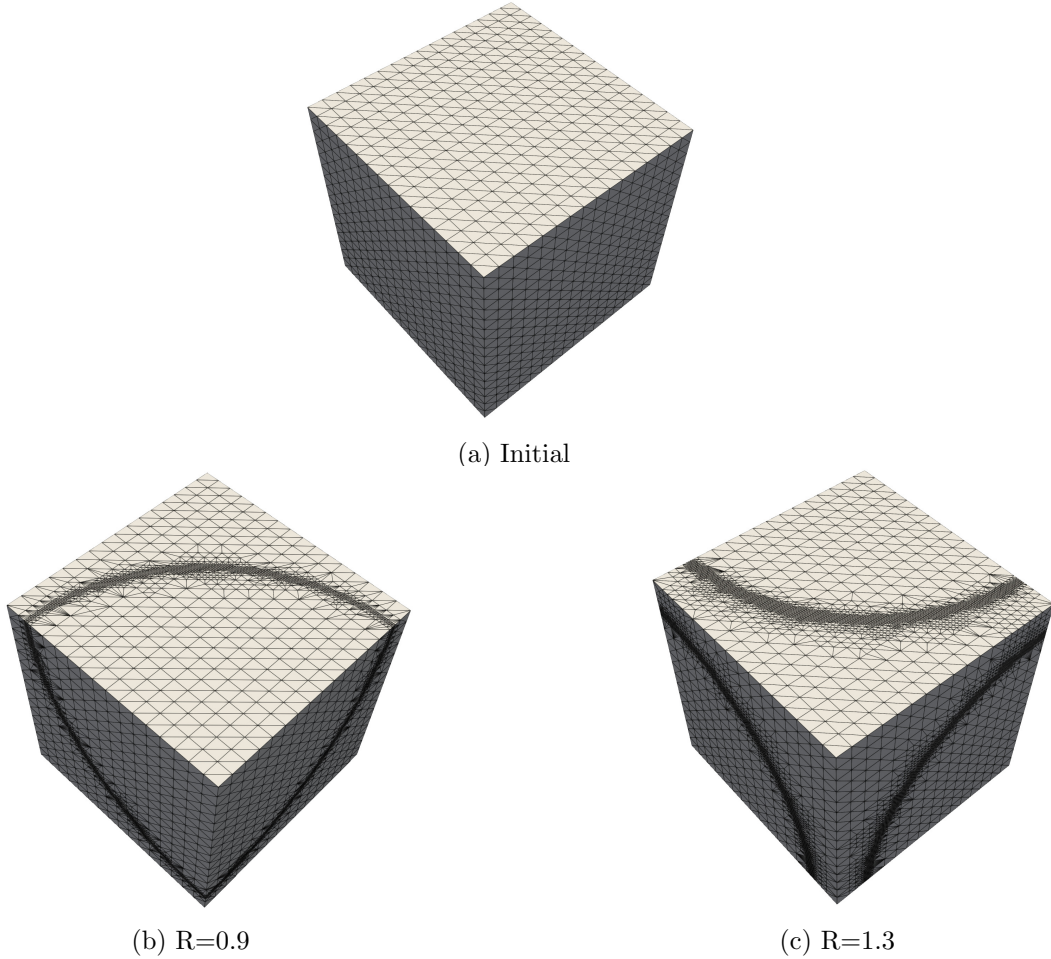


Figure 8.11: Three level h-adaptation in 3D. Note the conformal mesh propagation with alternative maps and the high valence nodes due to this propagation.

In conclusion, we have shown our novel for Nektar++ and NekMesh high-order isoparametric h -adaptation directly applicable for complex geometries. The 2D implementation does provide all three operators - split, swap, smooth and hence delivers all previously described benefits - preserving geometrical accuracy, improving mesh quality and providing good performance scaling. The 3D adaptation, on the other hand, does provide high-order conformal splitting and smoothing and does improve the overall mesh quality when used in an isotropic manner but will also benefit from high-order swap and parallelism. Therefore, for the purposes of this thesis, the *a posteriori* AMR applications will be limited to steady or quasi-unsteady h -adaptation.

Chapter 9

Towards AMR for Highly Separated Incompressible Flows

After confirming the 3D h-adaptation has been validated with analytical examples, it is possible to proceed with *a posteriori* AMR. Considering the complex industrial examples from part I, such as the full IFW, one can distinguish two main phenomena of interest - transition to turbulence and vortical structures development around the body and their interaction downstream, see Figure 7.1. Consequently, we will provide a proof-of-concept for such flows with feature-based error indicators and static h-adaptation.

The process works as follows: first, an initial solution is obtained on a coarse mesh with the Incompressible N-S solver of Nektar++ and the mean volume fields are used to evaluate a feature-based sensor that marks the elements to be adapted. Based on this sensor, the initial goal is ensuring sufficient streamwise x^+ and spanwise z^+ resolution through the h-adaptation, keeping the isoparametric prism layer splitting for the y^+ control, as in Part I. The extension to y^+ resolution control with variable prism layers is left for future work.

Performing AMR on the full IFW, however, is deemed too computationally expensive, with the non-refined cases requiring a Tier 1 HPC system. Therefore, the focus is on the extruded IFW from Part I as preparation for capturing the transition on the full IFW. To facilitate direct comparisons, the same numerical setup was used as by Liosi & Wustenberg and the Linear Implicit Time Integration Scheme with $\Delta t = 1.25 \times 10^{-4}$ s for these simulations [53, 111].

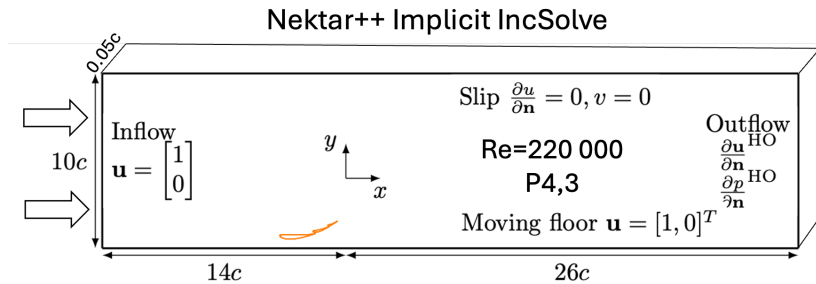


Figure 9.1: Numerical setup of the extruded IFW for the Nektar++ Incompressible N-S Solver for this thesis and [80, 111]

9.1 The choice of Feature-Based Error Indicators

While the error indicator is not the focus of this thesis, the author has explored the use of several feature-based sensors that are widely adopted in industry for incompressible aerodynamics. Namely, the total pressure ($|P_0|$), the Q -criterion, and the λ_2 , each providing different insights into the underlying physics.

While it is clear that the vortical indicators will show well the vortical regions, this is not necessarily the case for the transition structures. Therefore, to investigate that, we apply the sensors on the mean fields of the extruded IFW, which poses some challenges in resolving the transition to turbulence of the inverted multi-element wing. This analysis is based on an underresolved mesh, simulated for 14 Convective Time Units (CTU) and then time-averaged over the final 4 CTUs, as part of the early remeshing activities for capturing transition in the SSECOID project.

Starting from the total pressure (P_0) with a threshold value of 0.5, it does mark the region along the suction side of the inverted wing surfaces and throughout the wake region. This outcome aligns with the physical interpretation of the sensor - regions with an energy deficit indicated by reduced total pressure, typically coincides with highly separated and turbulent flow zones and wakes. In this case the mesh undergoes a considerable increase in mesh resolution, elevating the total element count from approximately 219×10^3 to 350×10^3 . This refinement is illustrated in Figure 9.2. However, what will stop us from using this indicator is the overrefinement in the wake of the wing. This refinement region extends from the main-plane up to the outlet, the whole 26 chord lengths. In the case of our motorsport applications, this is too excessive and the interest is on identification and control of vortices rather than the wake.

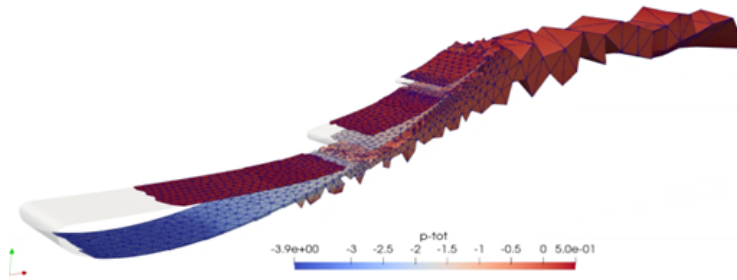


Figure 9.2: A region marked for adaptation based on the time-averaged over 4 CTUs total pressure (P_0) as the feature-based indicator. Note that the downstream refinement spans from the body to the end of the domain.

In contrast, the use of vortex-related sensors, such as the Q -criterion and λ_2 , marks notably different refinement zone. Applying the Q -criterion with a threshold of 5 primarily targets mesh refinement in regions near the wing's suction surfaces and downstream wake. The λ_2 criterion, set at a threshold of 100, identifies additional areas upstream and near the leading

edges of the wings while neglecting much of the downstream wake. Consequently, the λ_2 criterion incurs a higher computational cost, marking approximately 124×10^3 elements for refinement—approximately threefold compared to the Q -criterion. The differences between these two approaches are visually depicted in Figure 9.3, pointing towards the use of the Q -criterion for the flows of interest.

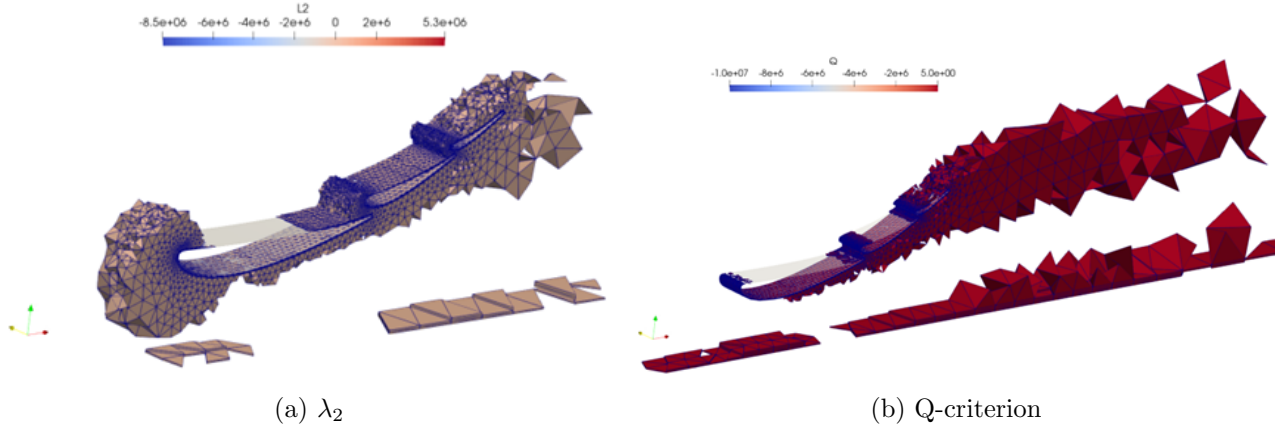


Figure 9.3: Elements marked for mesh adaptation based on time-averaged over 4 CTUs Q -criterion (> 5) and λ_2 -criterion (> 100).

9.2 Towards *a posteriori* AMR of the extruded Imperial Front Wing

The main zone of interest of the eIFW configuration is the Laminar Separation Bubble (LSB) forming on the suction side of the mainplane, visible in the mean fields of the simulations. This presented a challenging physical phenomenon requiring a full simulation campaign by Liosi & Wustenberg [111] with multiple different x^+ , y^+ configurations. At the moment of starting the AMR study, the y^+ requirements were established, whereas the x^+ was deemed to be a focus of the manual remeshing investigations for capturing the previously reported LSB by Slaughter [102] with Nektar++ 2.5D LES. Hence, the decision was taken to validate the proposed h-adaptation in terms of purely x^+ control while fully reusing the thickness, progression ratio and the number of prism layers on the body and the floor for the y^+ resolution.

The *a posteriori* AMR validation of the proposed h-adaptation works as follows. First, a very coarse mesh with good geometrical accuracy is created following the methodology of Part I. The same O-type macro prism layer settings were used as Wustenberg, and it was later split into seven prism layers, resulting in only 42×10^3 elements for the initial mesh. Then, the case was simulated until steady state was achieved and the mean field, over the last 4 CTUs, was used for the Q -criterion. Based on the Q -criterion user-defined threshold, the mesh was adapted

and the solution was interpolated onto the new mesh. This cycle was performed two times automatically (simulating each for 8-10CTUs and averaged over the last 4 CTUs). As the main interest of this case is the x^+ resolution, the surface meshes over the different cycles can be seen in Figure 9.4 a-c). In the first simulation, all three wings were marked for refinement, except the pressure side of the mainplane. This was also excluded in the later simulations, pushing the refinement zone towards the leading edge over the second and third h-adaptation cycles. This marks a significant difference from the reference cases, where a uniform mesh resolution was generated around the whole wing. Finally, as the LSB position becomes clear on the mean field, we also validated the manual refinement regions, adapting at and downstream the LSB achieving similar resolution at the LSB region as the most refined case by Wustenberg [111].



Figure 9.4: The high-order surface mesh of the initial, h-adapted and hh-adapted hybrid meshes based on the time-averaged Q-criterion. Please note the uniform refinement on the suction side of the wings and the lack of refinement on the pressure side of the mainplane and the first flap.

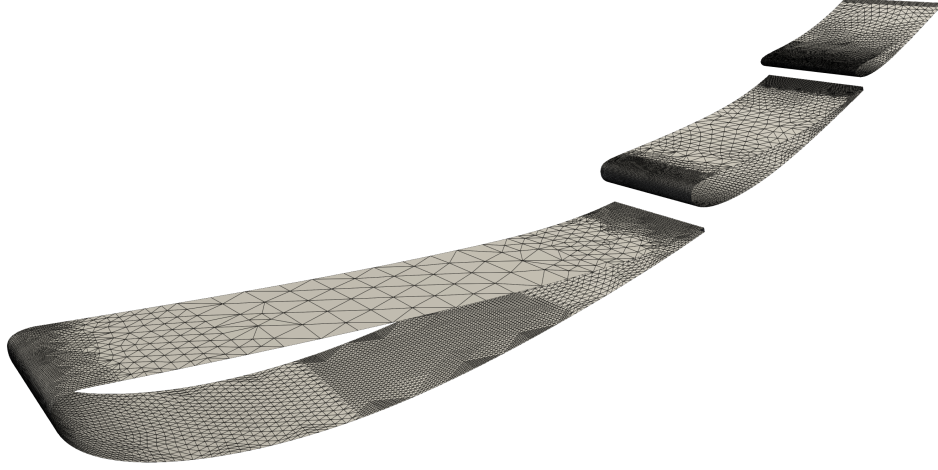


Figure 9.5: The high-order surface mesh showing the hhh-adapted mesh with manually selected region for refinement around and downstream the LSB.

Table 9.1: Overview of the AMR in terms of mesh size, mesh increase and an estimate for the computational cost per Δt_i (last 4 CTU) on HX1 8×64 Intel-Xeon P8358 CPUs.

Mesh	Mesh Size	↑ %	Driver	Δt_i [s]	Cost/ Δt_i	Comment
Initial	42×10^3	–	Q-auto	4×10^{-4}	2s	Strict AbsTol
h	190×10^3	450%	Q-auto	2×10^{-4}	5s	Strict AbsTol
hh	818×10^3	431%	Q-auto	2×10^{-4}	15.4s	Relaxed AbsTol
hhh	864×10^3	5.6%	Manual	1×10^{-4}	12s	10×64 CPUs Relaxed AbsTol

To analyze the physics after the AMR, one can observe the C_p distribution over the mainplane. For that, the mean-fields was calculated over at least 3 CTUs and spanwise slices are plotted in Figure 9.6. The initial mesh, in red, does not show any transition on the mainplane, which is expected due to the full mesh being only 42×10^3 elements, 3.8% of the reference. However, the extremely coarse mesh was sufficient to resolve the pressure side and to trigger adaptation on the suction side. The h-adapted mesh, in black, does start to capture a transition mechanism, but it exhibits an unphysical pressure rise at the laminar separation onset point of the LSB. A very similar region for adaptation is marked on the wings, but the pressure side of the second wing is also omitted from refinement. After the second hh-adapted simulation, one can already observe a clear pressure plateau inside the "dead-flow region" of the separation bubble, followed by a smooth increase in the pressure deficit pointing towards a separated turbulent shear layer and then a smooth reattachment. Some details about the simulations such as mesh sizes, relative increase and computational cost per Δt are summarised in Table 9.1.

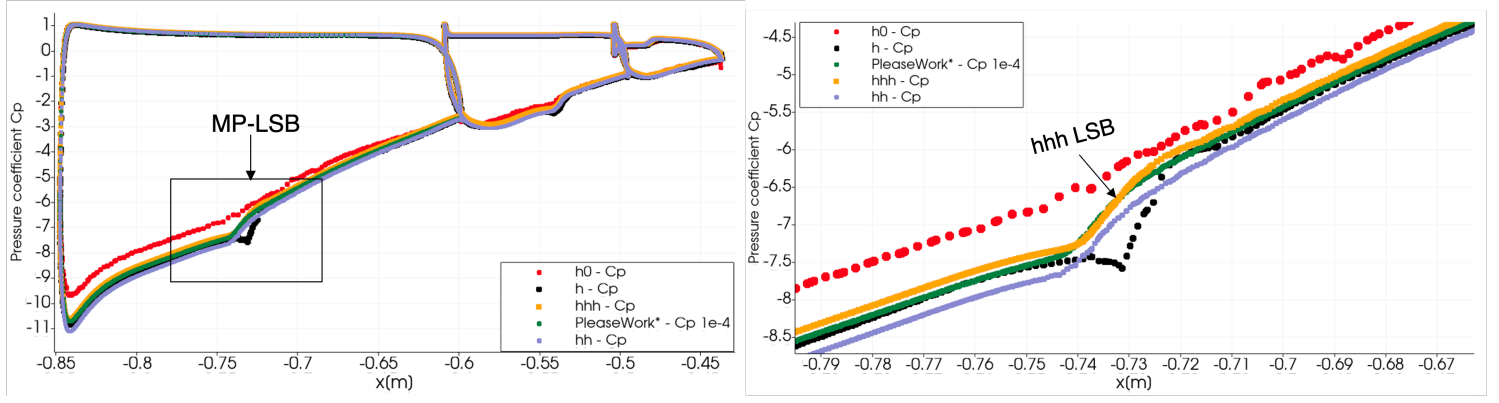


Figure 9.6: Time-averaged pressure coefficient (C_p) distributions on a mid-plane slice of the extruded Imperial Front Wing, comparing the initial mesh (red) and hhh-refined mesh (yellow) matching the same numerical setup with $\Delta t = 1 \times 10^{-4}$ s by Wustenberg (green), but manual remeshing (1.1×10^6 elements).

If one observes only the C_p and does not possess a more refined reference case, this would likely be the end of the AMR study, as the transition exhibits the expected pressure distribution. This suggests that the proposed *a posteriori* AMR is capable of resolving the favourable and the adverse pressure gradients even with a simple feature-based sensor as the Q-criterion, validating the h-adaptation as a process.

Last, by refining around the LSB region, it is possible to match the pressure distribution of our numerical reference by Wustenberg, who uses 1.1×10^6 elements, compared to the hhh-adapted being 8.6×10^5 , which is 21% less DoF. Further numerical gains could be achieved in the future by enforcing a minimum edge length estimated from x^+ , z^+ [91] or from geometrical local stopping criteria, since in the current simulation the resolution at the leading and trailing edges of the wing flaps exceeds the resolution of our finest numerical reference.

A more careful analysis of the viscous effects, however, shows that the second refinement (hh) mesh resolution is not sufficient. This can be seen through analysing the shear stresses or the skin friction coefficient $|C_f|$ on the same slices, see Figure 9.7. Hence, we perform the last manual refinement, yellow, which matches the numerical reference by Wustenberg upstream of the LSB fully in C_p . It also resolves near identically the upstream $|C_f|$, delaying slightly the laminar separation point, but resolving identically the "dead-flow region" in the LSB and the maximum shear stress in the region of reattachment. Downstream, it does continue to match the reference until the end of the refinement region. Then, when the flow reaches the coarser mesh, the larger elements act as a spatial filter, smoothing the finer flow structures and consequently the solution underestimates the skin friction coefficient.

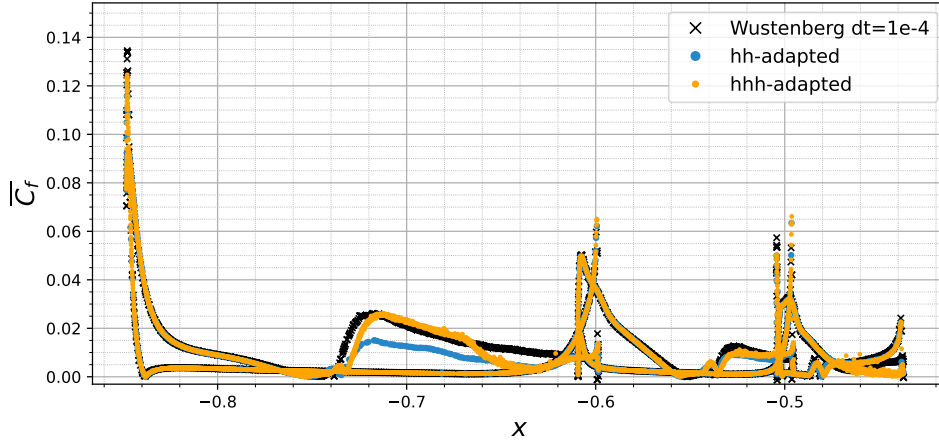


Figure 9.7: Time-averaged skin friction coefficient (C_f) distributions on slices of the extruded Imperial Front Wing, comparing the hhh-refined mesh (black) with the same numerical setup with $\Delta t = 1 \times 10^{-4}$ s by Wustenberg, but uniformly remeshed with 1.1M elements. Note the good agreement up to the reattachment region, and downstream up to the end of the manual refinement zone of the hhh-adapted mesh.

This last iteration was a manual refinement step, and therefore, one can extend this refinement region over the full mainplane, as the reference resolution over the wing flaps' suction sides has already been achieved during the second cycle of refinement. This results in an overall mesh count of 910×10^3 elements and hence an element count reduction of 17% compared to the one reported by Wustenberg [111]. A more rigorous approach would be to rely on the Q-criterion with a well-defined min edge length for another AMR cycle, but these simulations have been left for future work as the solution was deemed already sufficient for answering the initial research question.

Even as it is, the test case demonstrated that our AMR leads to an improved resolution in the areas of interest and does capture the laminar separation bubble on the main plane automatically. Moreover, this is achieved without a sophisticated adaptation driver with a suboptimal feature sensor and only $x+$ control. Hence, even achieving this 17% decrease in element count presents a milestone for our proof-of-concept 3D AMR in Nektar++ with the high-order h-adaptation - starting from a very coarse, under-resolved simulation.

Chapter 10

Towards h-r-p adaptation

The final chapter of this thesis will investigate the promise for the full potential of hp/SEM methods in combining the best from the spectral convergence for smooth flows and the flexibility and efficiency of h,r-adaptation. This will lead to a proof-of-concept h-r-p AMR, where we leverage the previous experience in the research group of r and p-adaptation [61, 27]. This study focuses on compressible flows (inviscid), where both smooth and discontinuous features, such as shocks, are present, making them an ideal test case for h-r-p adaptation that is also sensitive to geometrical perturbations. Moreover, we explore the different AMR configurations and propose the most efficient one. Since one can expect tens of multilevel adaptation configurations, we will limit this study to 2D due to the computational cost.

We also extend our answer to the second research question - could the h-adaptation indeed resolve the flow features found in compressible flows, such as shocks and interfaces, starting from very coarse underresolved meshes outside of the asymptotic region.

Hence, first we will introduce some of the main components of the framework such as the Governing Equations and the shock sensor, followed by the r and p-adaptation techniques. Then we will demonstrate the AMR with a Supersonic intake for investigating the pros and cons of the h, r mesh modification techniques for capturing the shocks. Last, we will follow up with the final recommendations for attempting to define an efficient strategy for h-r-p mesh adaptation for a transonic steady NACA0012 case featuring a curved geometry.

10.1 Governing Equations

The (inviscid) compressible flows are described by the Euler Governing Equations. We can write the Euler Equations as the sum of the time derivative of the conserved variables $\vec{u}$ and the divergence of the flux vector $\mathbf{F}(\vec{u})$, constructed from the convective $\mathbf{F}_c$ and artificial dissipative $\mathbf{F}_d$ fluxes, used by the Nektar++ DG solver to stabilize the solution.

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot [\mathbf{F}_c(\vec{u}) + \mathbf{F}_d(\vec{u}, \nabla \vec{u})] = 0. \quad (10.1)$$

The conserved variables $\vec{u}$ and the convective flux $\mathbf{F}_c$ for the two dimensional Euler Equations are:

$$\vec{u} = \begin{Bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho E \end{Bmatrix}, \quad \mathbf{F}_c(\vec{u}) = \begin{bmatrix} \rho v_x & \rho v_y \\ \rho v_x^2 + P & \rho v_x v_y \\ \rho v_x v_y & \rho v_y^2 + P \\ \rho v_x H & \rho v_y H \end{bmatrix}, \quad (10.2)$$

The system of equations is closed with the total enthalpy H via perfect gas law, linking the pressure P with the energy E , where we choose a standard air specific heat ratio γ .

$$H = E + \frac{P}{\rho}, \quad P = (\gamma - 1)\rho \left(E - \frac{v_x^2 + v_y^2}{2} \right), \quad \gamma = 1.4. \quad (10.3)$$

Throughout this chapter we use slip-wall boundary condition on the body with the no permeability condition, where the accurate curved geometry becomes important. For the far-field boundary conditions, the free stream conservative variables are imposed through the Riemann solver.

$$\vec{v} \cdot \vec{n} = 0 \quad \text{on } \Gamma_{\text{wall}} \quad (10.4)$$

10.2 Shock Capturing and Smoothness Sensor

The dissipative flux introduced in Equation 10.1 does not appear in the original Euler equations, but is crucial for stabilizing the flow, especially around discontinuities and shocks featuring the Gibbs phenomenon. Our Nektar++ implementation uses the method proposed by Persson and Peraire [86], where the dissipative flux is applied with a constant artificial viscosity μ_a^e across each element.

$$\mathbf{F}_d^e = -\mu_a^e(\vec{u}_h^e) \nabla \vec{u}_h^e. \quad (10.5)$$

Please note the term is a function of h^e elemental size, p^e the order, μ_0 is a user-dependent scaling factor, $\lambda_{\max}$ the maximum wave speed of the system, and $\mathcal{S}^e$ is where the shock capturing appears.

$$\mu_a^e(\vec{u}_h^e) = \mu_0 \frac{h^e}{p^e} \lambda_{\max} \mathcal{S}^e(\vec{u}_h^e), \quad (10.6)$$

To calculate the $\mathcal{S}^e$, a modal smoothness sensor is defined as per Persson and Peraire, where the discontinuities are estimated as the normalized difference between the conserved variable q at two polynomial orders. In our examples we choose ρ as for this state variable.

$$s^e = \log_{10} \left(\frac{\langle q - \tilde{q}, q - \tilde{q} \rangle}{\langle q, q \rangle} \right). \quad (10.7)$$

Where q and $\tilde{q}$ are the full and truncated expansions of the state solution. As we use hierarchical modal basis, this is available at any point of the solution.

At this point we can define $\mathcal{S}^e(\vec{u}_h^e)$. To apply the artificial viscosity only when discontinuities arise a smoothness function is introduced, splitting it into three dissipation regions with the help of user defined κ parameter.

$$\mathcal{S}^e(\vec{u}_h^e) = \begin{cases} 0 & \text{if } s^e < s_0^e - \kappa \\ \frac{1}{2} \left[1 + \sin \left(\frac{\pi(s^e(\vec{u}_h^e) - s_0^e)}{2\kappa} \right) \right] & \text{if } |s^e - s_0^e| \leq \kappa \\ 1 & \text{if } s^e > s_0^e + \kappa \end{cases} \quad (10.8)$$

where $s_0 \sim \log_{10}(p^4)$ from an analogy to Fourier coefficients decaying as $1/p^2$ and s_κ a user defined parameter controlling the cut-off region values.

$$s_0^e = s_\kappa - 4.25 \log_{10}(p^e), \quad (10.9)$$

Please note that throughout this work, we will refer the smoothness sensor as the PP-sensor and will use it to drive the AMR to refine the resolution in regions of interest.

10.3 R-adaptation

The r-adaptation adopted in this work achieves the resolution control through mesh movement while keeping the overall mesh topology (DOF and N_{el}) constant. To achieve that, we use the variational framework presented in part I, as per Marcon and Turner [61]. This approach allows mesh movement both for the linear straight-sided mesh as per Marcon [61] or directly to the curved high-order one.

Variational Framework The core of the r-adaptation is the variational optimization framework that creates a higher level abstraction combining the classical PDE-based and deformation energy approaches into a pointwise single formulation, as per Turner [106]. This approach can be extended to the full mesh by solving multiple small optimization problems for minimizing

the energy functional defined on the elements over the mesh. In this case, we use the same hyperelastic formulation as for the volume optimizer in Section 3.4.2, please recall Equation 10.11.

$$\text{find } \min_{\boldsymbol{\varphi}} \mathcal{E}(\nabla \boldsymbol{\varphi}) = \sum_e \int_{\Omega^e} W(\nabla \boldsymbol{\varphi}) d\mathbf{y}, \quad (10.10)$$

$$W = \frac{\mu}{2} (I_1^C - 3) - \mu \ln J + \frac{\lambda}{2} (\ln J)^2, \quad (10.11)$$

R-adaptation To perform r-adaptation, we follow the isoparametric mapping notation from physical to reference element and backwards. However, the mapping ϕ is extended by defining an intermediate ideal or target element Ω_T^e . In this case, we can take the current ideal element, modify the map ϕ_T with a metric tensor $\mathbf{M}$ and with that, create a new target ideal element, which will then be used in the energy formulation and will guide the variational optimiser. In this study, we apply only the isotropic r-adaptation, which can simplify the Metric tensor to Equation 10.13. The only parameter required for the adaptation then is the scale $r = h_T/h_I$ that is the ratio between the target and the current ideal element and is passed to every high-order node. This is then used over the mesh movement iterations.

$$\mathbf{J}_T = \mathbf{M} \mathbf{J}_I = (r \mathbf{I}) \mathbf{J}_I \quad (10.12)$$

$$\mathbf{M} = \begin{bmatrix} \frac{1}{h^2} & \\ & \frac{1}{h^2} \end{bmatrix} \quad (10.13)$$

An important part of the variational r-adaptation is the node sliding on the CAD surfaces. Since we combine that with h-adaptation, the mesh-CAD connectivity will be recovered using the CAD-reconstruction from Part I of the thesis, allowing for the crucial increase in the resolution on the body itself.

10.4 P-adaptation

The variable polynomial order compressible DG flow solver of Nektar++ allows for resolution control through p-adaptation, while ensuring the flux continuation with the Riemann solver on the elemental interfaces. Different combinations for the p-adaptation were tried during this thesis - for example limiting the p around discontinuities to p_{min} or directly jumping to the allowed p_{max} in a single adaptation. However, as this is a proof-of-concept focusing on the

combination of mesh modifications, we opt for the classical approach where a local change of a single polynomial order will be applied for a sensor threshold. Moreover, as the anisotropy has been reported to be obtained even with the isotropic r-adaptation [61], we keep the p-adaptation to an isotropic as per the initial implementation by Ekelschot [27]. The process is described in Algorithm 5 and works as follows: First, the solution u^n is obtained and the elemental Persson-Peraire (PP) sensor $\mu_e(u)$ is calculated [86]. Then, based on user-defined threshold of $\mu_e(u)$, the new elemental polynomial order is defined as p_e . Last, for the elements with modified p_e , the basis function is updated to accommodate the new order and the solution is projected onto the new basis, making the process ready for the next adaptation cycle.

Algorithm 5 Isotropic p-Adaptation algorithm

- **Input:** Element-wise Persson-Peraire PP-smoothness indicator $\mu_e(u)$
 - **Output:** Updated Expansion with Projected Solution
1. **For each element e in the mesh:**
 - If $\mu_e > \mu_{up}$, set $p_e = p+1$ (shock detected)
 - Else if $\mu_{low} \leq \mu_e \leq \mu_{up}$, set $p_e = p+1$ (smooth)
 - Else, set $p_e = p_0$ (low error)
 2. **Update basis:** Updated basis functions ϕ_i^e using new p_e
 3. **Project solution:** Project current solution $\hat{u}^e(x)$ onto new basis
-

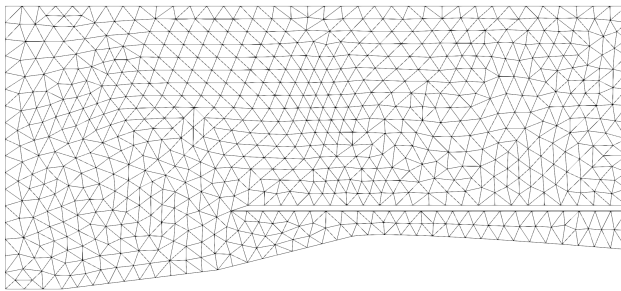
At this point, we can proceed with combining the existing AMR techniques and the r and p-adaptation previously developed in the group with the novel h-adaptation we have presented. We will start from the purely shock-capturing example of a supersonic intake and then finalize Part II on examining the benefits of a proof-of-concept combination of h-r-p adaptation with an inviscid transonic NACA0012 example.

10.5 From h to hr - Supersonic intake

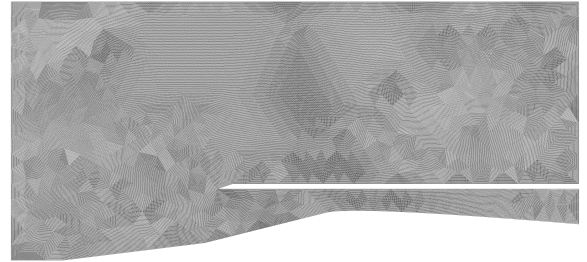
The first application is a supersonic intake as described by Jain [40] and used by Marcon [61]. The geometry starts with an intake having a uniform flow at Mach 3.0, which is then deflected by the two straight-sided inclined ramps at 7° and 14° angles, respectively. Then, the diffuser of the engine is represented by a curved NURBS. The horizontal wall on the top surface of the intake is straight-sided but has a carefully positioned sharp leading edge precisely on the way of the oblique shockwave created from the first ramp. This results in a reflection on the horizontal surface and complex oblique shock patterns with multiple reflections in a small confined domain and shocks with different strengths. Hence, this is a perfect test case for the capability and the limitations of the different adaptation techniques for capturing shocks with different strength, their complex interactions and reflections to curved walls. As we will be focusing on capturing the shocks, our initial tests show that low polynomial orders are sufficient for shock capturing and are most computationally efficient, while not completely sacrificing the curved geometry. Therefore, we will continue with $p=2$ over this example.

10.5.1 Uniform Refinement and Reference Solution

As a starting point, we create an initial very coarse base mesh and refine it uniformly four times to create a reference case for numerical comparison and verification. For this, a very coarse uniform triangular $p = 4$ mesh is created with a mean size $\Delta h = 26 \times 10^{-3}$ and $N_{el} = 1410$ triangles, see Figure 10.1 a).



(a) Initial ($N_{el}=1410$)



(b) Reference ($N_{el}=360 \times 10^3$)

Figure 10.1: Comparison between the initial $p=4$ mesh used to start all adaptations (a) and four level uniformly refined mesh that is used as a reference "reference" solution (b)

Then, uniform refinement is applied four times with the finest 1:4 map. Every case is simulated, initialized from the previous refinement level with the Nektar++ Euler Solver $p=2$. The finest mesh, $N_{el}=360 \times 10^3$ is used through this study as a reference solution u_e . This reference solution can be seen in Figure 10.2. Please note that inside the intake, the case consists of two separate oblique shocks and their reflection on the curved surfaces results in an expansion fan that, most

importantly, is vastly weaker compared to the main shock. This can be observed more clearly using the Persson-Peraire(PP) shock-capturing sensor, see Figure 10.3. We use this sensor both for the artificial dissipation and for driving the adaptation. The final simulation took 3.8h (128CPUs) for the finest mesh ($t_{end} = 0.1s$) and explicit time stepping $\Delta t = 9.2 \times 10^{-8}s$.

The reference solution clearly demonstrates the main challenge for the AMR process - how to locate both the strong and the weak shocks correctly. And while the strong ones have already been shown to be possible to be captured with rp-adaptation by Marcon, the author is not aware of an AMR that has successfully shown the secondary weaker shocks and the expansion fan on the curved surface. Therefore, if successful in starting from the base mesh that has 257 times fewer DoF and only two elements between the intake walls, this will unambiguously prove the second goal of Part II. As the goal is only capturing the shocks, we limit the investigations to h and r-adaptation, starting from our h-adaptation and then continuing with r and hr-adaptations. In this case, we will use instantaneous values of the PP-sensor to drive the adaptation as an attempt to get it closer to future dynamic AMR.

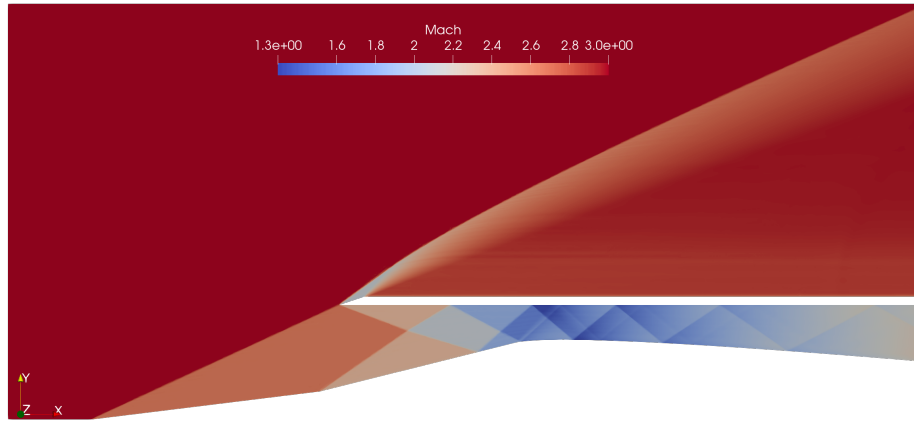


Figure 10.2: Supersonic Intake at Mach 3 for the reference u_e . Note, inside the intake a strong and weak shocks develop, reflect and propagate, both reaching the outlet.

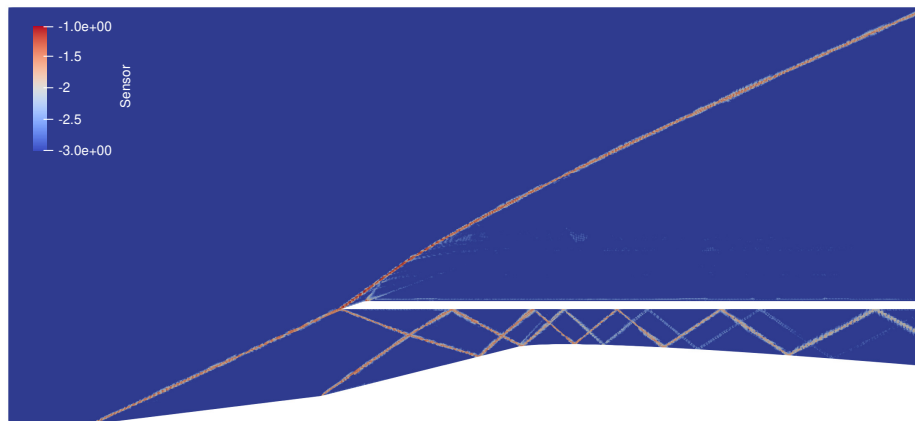


Figure 10.3: Persson-Peraire shock sensor. Note the strong and weak shocks develop, reflect and propagate, inside the intake both reaching the outlet. PP illustrated $[-1; -3]$

10.5.2 h-adaptation

The h-adapted case works as follows. We start from the initial base mesh and initialize the solver from uniform free-stream conditions and simulate until a steady state is reached. Then, based on the PP smoothness indicator, the h-adaptation is applied with 1:4 refinement map to the elements marked for being inside the shock region. This is followed by swap, two smoothings ($\alpha = 0.35$) and a final swap operation, performed on the adapted mesh. The previous solution is interpolated to the new mesh and then simulated further for a minimum 2 CTU (c_{domain}) with more up to h_3 to flush out any oscillations and develop the flow, followed by another h-adaptation cycle. It is important to note that we impose a minimum target edge length after the splitting to be $h_{min} = 2 \times 10^{-3}$, which limits the adaptation up to the edge lengths of the reference mesh. This process is repeated until more than 85% of the marked for adaptation elements are below the user-defined h_{min} as a user-defined stopping criterion, pointing towards reaching a "converged" AMR state. In this case, this is achieved in five h-adaptation cycles. The evolution of the meshes and the solution in terms of PP-sensor can be seen in Figure 10.4.

The solution progression looks as follows. First, we use the h_0 case to reach the initial steady state. As this base mesh has 257 times fewer DoF compared to the reference case, this is near instantaneous and even though the solution is unusable from physics perspectives inside the intake, the PP-still marks the regions of interest for adaptation (the intake), but not upstream of the first oblique shock. Then the h_1 and h_2 both capture the location of the stronger shock and its reflections internally in the intake, but also an expansion wave that develops outside/top of the engine LE. In h_3 we also start to observe the weaker shock that reflects on the bottom surface of the intake throat, but this is spatially filtered downstream. This is improved by h_4 , which reaches the minimum desired size h_i in the majority of the strong shock regions, but does not fully resolve the weaker shock. Finally, h_5 ($N_{el} = 36 \times 10^3$) does propagate the weaker shock refinement downstream the intake, almost reaching the outlet, by adding only 12.5% elements on top of h_4 ($N_{el} = 32 \times 10^3$). It is crucial to mention the use of a minimum size h_{min} similar to the reference mesh, as if not applied, the mesh size would have doubled in the region of the stronger shock for h_5 and the timestep Δt_i halved, hindering the performance gains. The h_5 mesh qualitatively resolves all shocks inside and outside the intake and, in fact, reaches similar resolution as our reference mesh in the shocks, while using only 10% of the DoF and the last simulation being x8 times faster than the uniform.

Another well-refined zone in the simulation is the expansion fan that forms outside the intake, especially downstream of the main shock. While also physically evident in the reference uniform solution, this is overrefined in h-adapted meshes. We explain that with the use of the instantaneous solutions for the PP-sensor and the very low dissipation during the AMR cycling, hence numerical artefacts are propagated through the domain - an undesirable overrefinement, but an expected effect in a future dynamic adaptation case for feature-based AMR.

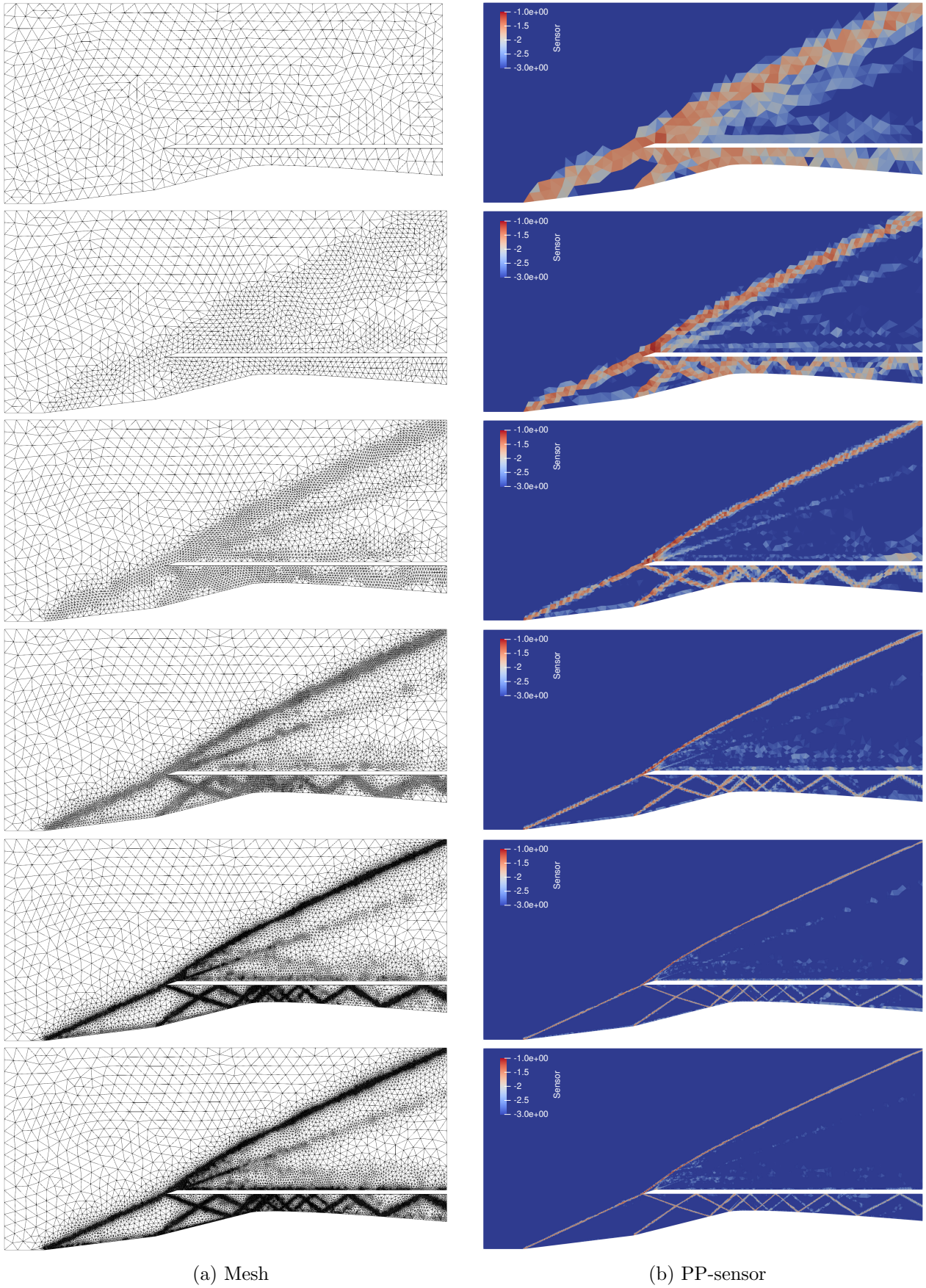


Figure 10.4: Mesh and the PP indicator through the AMR evolution. Please note the indicator is limited to $[-3, -1]$ for approximate visualization purposes of what elements are marked for refinement.

10.5.3 r and rp -adaptation

Before moving to the final goal of this application, the hr -AMR, it is worth discussing the work by Marcon on r and p -adaptation with the same solver [61]. This test case is a great demonstrator for the use of the r -adaptation due to the very confined region inside the intake and the multiple shocks inside. Based on his findings and our independent analysis, we concluded that pure r or rp -adaptation is not sufficient to resolve the shocks for the very coarse initial mesh. This is consistent with the findings of Marcon [61]. The rp -adaptation, in his case, managed to resolve very well the first strong shock after starting with $N_{el}=4021$, carefully placed a priori in the region of interest. In this case, the mesh in the main shock appeared highly anisotropic and aligned with the shock, but even with that, only the first shock reached the expected resolution and the rest was coarser than h_2 , see Figure 10.5. Hence, despite the simulation converging to a state with low values of the PP-sensor, it fails to resolve the weak shock at all due to the spatial filtering. One might suggest a continuous refinement with hr on this mesh, but the initial use of rr -mesh $\Delta t_i = O(10^{-9})$ s is prohibitive for any practical purpose for further refinement.

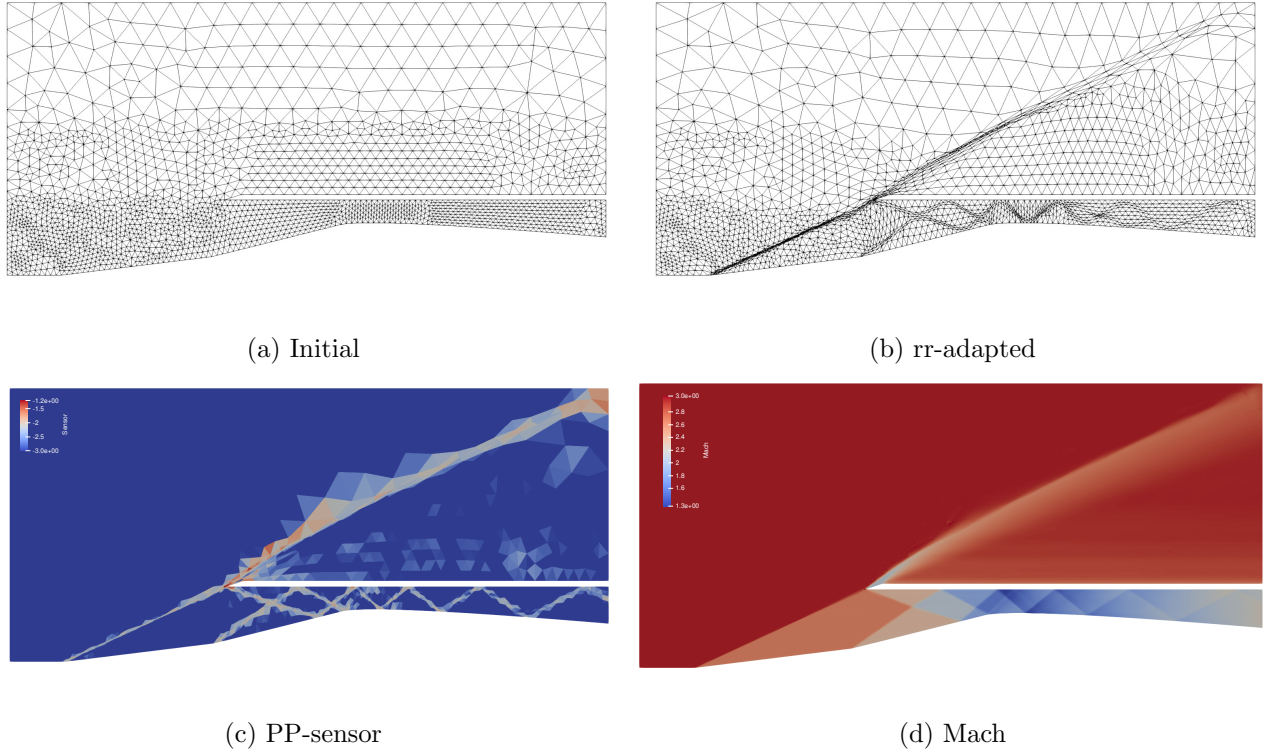


Figure 10.5: Mesh and PP error indicator for the rr -adapted mesh in the study by Marcon [61]

In conclusion, while r and rp -approaches can be considered extremely efficient due to keeping the connectivity and not introducing new DoF - they can only work if the initial mesh has already resolved correctly the location of the shocks and there are sufficient DoF around the shock regions that will accommodate the r -adaptation without deteriorating the weaker shocks and the overall accuracy.

10.5.4 hr-adaptation

A natural alleviation of the issues of the r-adaptation is its combination with the flexibility of the h-adaptation. And more precisely, employing the h-adaptation until both the weak and the strong shocks are well resolved, followed by an r-adaptation that aligns the mesh with the shocks anisotropically, drastically improving their resolution. If we assume only one r-adaptation cycle with $r_i=0.5$ scaling, this would translate to h-adaptation reaching h_{min} up to $2 \times h_{target}$. In pure h-adaptation, this would refer to the third AMR cycle, h_3 . Hence, this mesh will also serve as the starting point for the hr-adaptation. However, applying the r-adaptation straight away would be a mistake as the secondary weak shocks inside the intake are not visible downstream at this point and the PP-sensor has spread the adaptation target to a wide area, making the r-adaptation ineffective. A better approach is to allow further h-adaptation with this higher minimum size $h_{min}=4 \times 10^{-3}$ and the same $< 85\%$ of the marked elements below h_{min} as a stopping criteria. This adds another two h-adaptation cycles that propagate the refinement around the weaker shock downstream the engine diffuser, while keeping the same resolution for the strongest shocks upstream, see Figure 10.6 a). At this point of the adaptation, we reach the stopping criteria of the h-adaptation for min size and apply the r-adaptation. The resulting mesh and the PP-sensor can be seen in Figure 10.6 b).

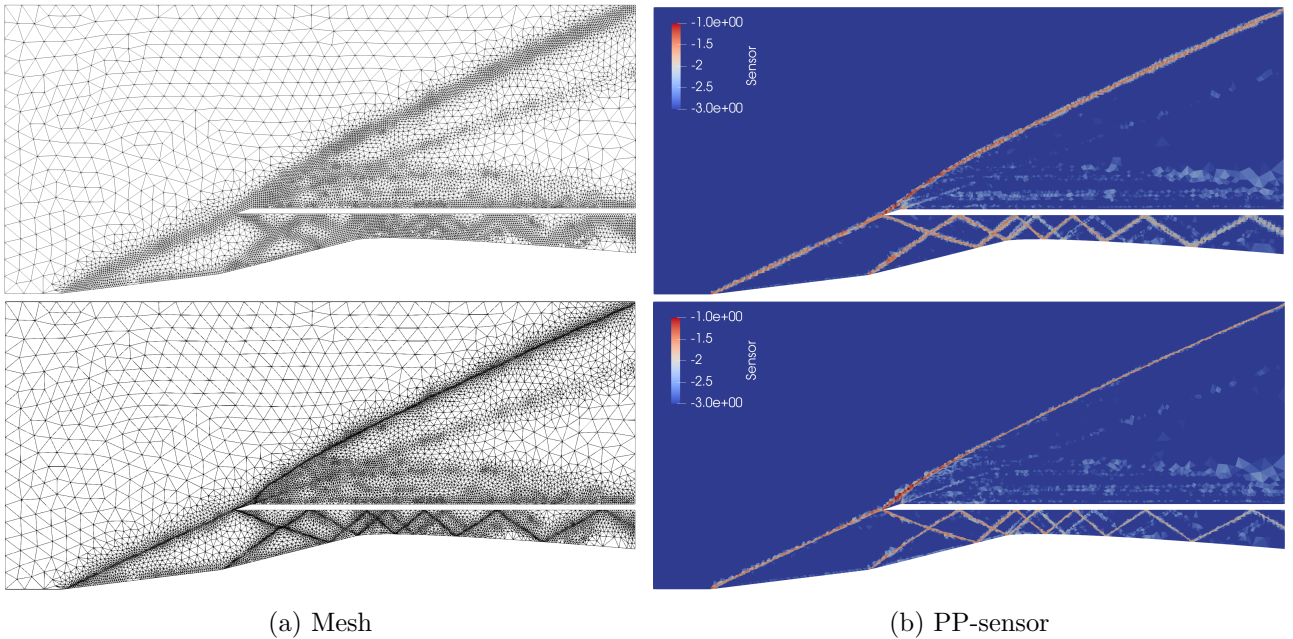


Figure 10.6: Mesh and PP-sensor for the h-adapted ($h_{min} = 4 \times 10^{-3}$) and the hr-adapted mesh and the PP-sensor for the solutions. Please note the highly resolved strong shocks and the improved weaker ones up to the fourth reflection of the weaker shock.

A close inspection of the mesh shows the highly anisotropic and well-refined stronger shocks with a finer mesh size than the reference solution. The mesh size, however, has higher size variability around the shocks, not fully reaching the h_{target} of the reference case close to the

outlet. The weaker shocks are excellently resolved up to the fourth reflection, while for the last two, they are not visibly refined and rely on the resolution from the *h*-refinement. From the PP-sensor, the stronger shocks are well resolved, with the exception of the expansion wave on top of the intake leading edge being slightly distorted. The weaker shocks are also well refined with up to the fourth reflection, resulting in qualitatively similar results as the pure *h*-adaptation, but at the cost $N_{el} = 1.9 \times 10^4$ elements (46%) fewer DoF and the same final Δt as the *h*-adaptation.

10.5.5 Shock Capturing and Accuracy

A more detailed examination is required for the location and the strengths of the shocks to determine if this cost-saving *h* and *hr*-AMR does not negatively impact the accuracy. To analyse that we rely on the Mach number of the top wall inside the engine as one of the main design criteria. An overall comparison between the *h*, *r* [61] and *hr* versus the reference solution can be seen in Figure 10.7.

From this, one can conclude that the strongest shocks around the inlet ($x < 0.55$) are very well resolved by all simulations, with a near perfect agreement for the *h* and *hr* with the reference. However, when looking downstream of the intake, substantial differences start appearing with the *r*-adaptation completely smoothing the second and third weaker shocks at $x=0.67$, $x=0.8$ and shifting the stronger shocks downstream compared to the reference. This was expected, as the weaker shocks were not captured by the *r*-solution at all, and hence their resolution was not improved during the *rr*-AMR in Marcon's work.

On the other hand, the *h*- and *hr*-solutions do resolve well all stronger shocks also downstream the intake throat, with the *h*-adaptation showing identical Mach as the reference with even the undershoot at one at $x=0.63$ being of the same shape. This can be expected as the mesh resolution on the shocks is the same between the reference and *h*-adaptation case, but also shows the equivalence between the AMR and reference solution around the inlet. The *hr* also shows a good agreement with the reference solution with minor overshoot $\Delta M = 2 \times 10^{-2}$ at $x=0.76$. All this shows that even furthest downstream, both the *h* and *hr*-adaptation are capable of capturing the locations of the shocks correctly and that the cost savings (46%) of the *hr*-adaptation does not negatively affect the stronger shocks. This makes the *r*-adaptation the operation of choice once the shocks are already identified by the *h*-adaptation and the PP-sensor.

Furthermore, the *h* and *hr*-adaptation do capture all weaker shocks in this complex configuration, which is remarkable given they use 10 and 19 times fewer DoF, respectively, compared to the reference case. For a more critical analysis and completeness, the second weaker shock is resolved well in both the *h* and *hr*, with maximum deviation of $\Delta M = 1 \times 10^{-2}$, but with

minor upwinding for the hr and slight undershoot of the h-AMR. A closer analysis of the most downstream weaker shocks does present a negligible wider front, suggesting more diffusive on the hr, while the h- moving upstream, but only $\Delta x O(10^{-3})$.

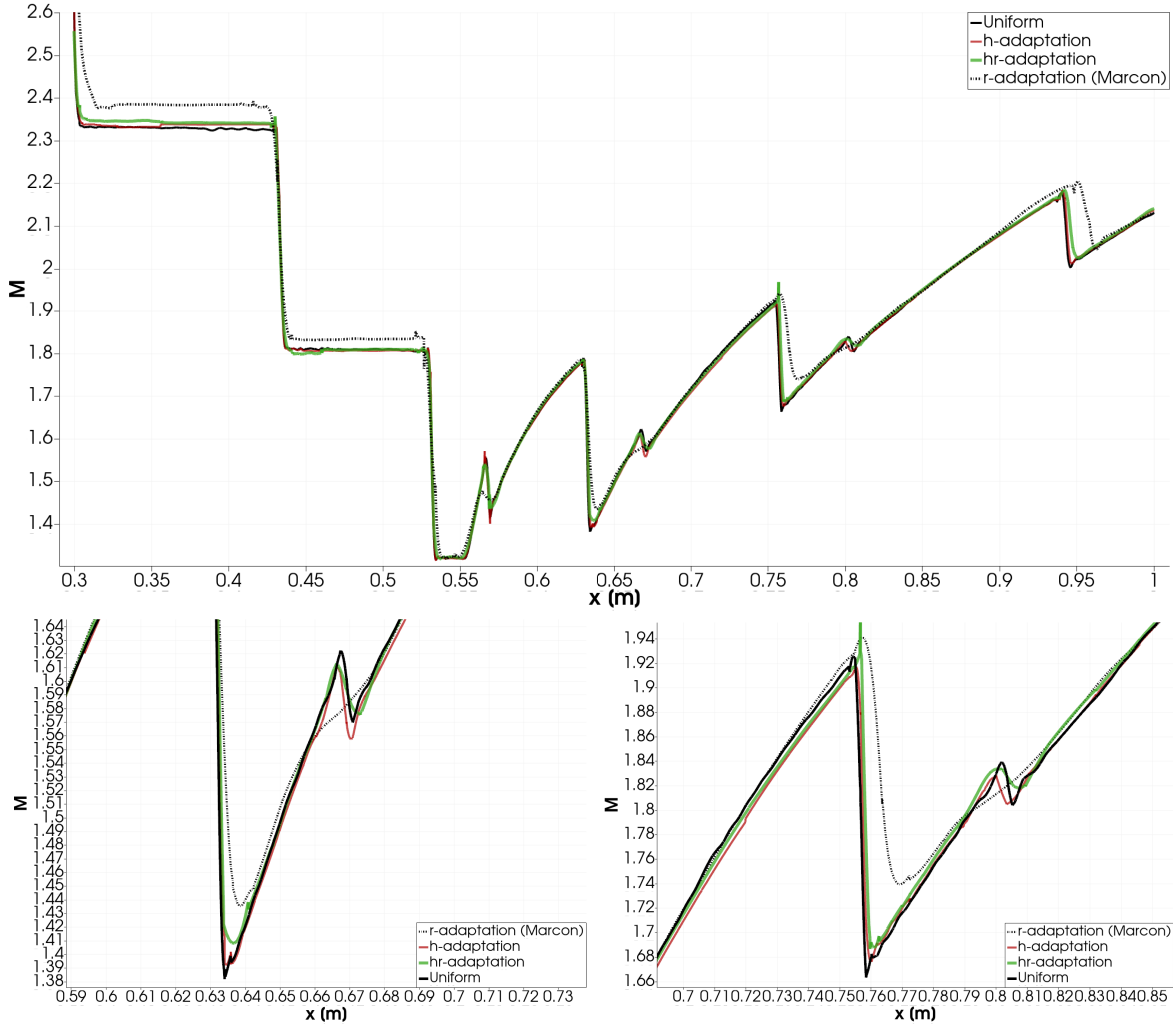


Figure 10.7: Comparison between the reference solution (Mach Number) with the h, r [61] and hr solutions on the full top wall of the intake and zoomed in the weaker shocks two and three. Please note that all shocks are captured with both h and hr-AMR, but not with r-adaptation.

Although it is important to acknowledge the differences relative to the uniform solution, the magnitude of these deviations confirms the capability of the h-adaptation in accurately resolving the location and strength of the shocks in the complex intake configuration. This was found to be true even when the case has shocks having different strengths and being close in confined space, all achieved even starting from very coarse meshes.

Last, the hr-adaptation is the configuration of choice for AMR, having the efficiency gains of the r- and the flexibility of the h-adaptation without exhibiting a significant accuracy penalty. An effective hr-adaptation will start with h-refinement to resolve the shocks and only when clearly distinguishable, will be followed by r-adaptation to improve the final resolution and achieve the anisotropy.

10.6 Full hrp-AMR - Transonic NACA0012

Having established the strategy for capturing shocks, we can proceed with the final goal of this part - assessing the capabilities and establishing a full h-r-p AMR. To verify our approach, we consider the flow around NACA0012 in a transonic regime. This test case has been widely used in literature due to the occurrence of a single strong shock on the suction side and a weak shock on the pressure side of the airflow as per Marcon [61], Balan [5], Vassberg and Jameson [108]. The complexity of the case comes from the curved geometry, especially on the Leading Edge (LE) of the airfoil and capturing the location of the shocks, given their different strengths and sensitivity to upstream perturbations. As this is an easier configuration for the shock capturing, we will use the airfoil for adding also the p-adaptation into the workflow, moving towards our proof-of-concept of full h-r-p framework.

The NACA0012 test case used in this example is at transonic flow conditions, $M = 0.8$. The airfoil is positioned at $AoA=1.25^\circ$, with chord $c=1m$, placed in the middle of a square domain with length $L=30c$. The Nektar++ Euler Solver is used with a slip wall boundary condition for the airfoil and a farfield free-stream boundary condition for the domain. The numerical setup is with third-order Explicit Runge Kutta time-stepping and HLLC are used for the Riemann solver. The artificial viscosity approach used here starts with a low artificial viscosity during the AMR-cycles ($\mu_0=0.02$) for easier shock detection and then, for the final mesh/simulation, increases ($\mu_0=0.06$) for smoothing out the shocks under and overshoots. This change comes from the use of the PP-sensor both for adaptation and for determining the amount of artificial dissipation added as a stabilization, which smooths out the shocks, potentially causing the elements with the weaker shocks to be indistinguishable from the smooth regions, especially in the early stages of the adaptation.

10.6.1 Initial Mesh, Uniform Refinement and Reference Solution

As for the intake case, we create a very coarse initial mesh, refer to Figure 10.8 $p=4$ mesh consisting of $N_{el}=858$ with NekMesh and the Variational Optimizer. We intentionally avoid creating a fully uniform mesh to introduce an additional challenge from an AMR perspective, which is more representative of a real-case scenario. For a reference mesh, we start from a similar initial mesh and apply h-refinement four times uniformly with the finest 1:4 map, resulting in $N_{el}=318 \times 10^3$. At $p=3$ this resulted in $N_{DoF}=3.18 \times 10^6$ and $\Delta t=5 \times 10^{-7}s$.

While the intake was kept at lower order, here the LE has high-curvature, hence we would run our reference case with $p=3$ and the global dealiasing $2 \times p$. The solution and the artificial viscosity can be seen in Figure 10.9. Please note, that no excessive artificial dissipation is applied outside of the shocks and the trailing edge. Moreover, higher than quadratic polynomial order is necessary to avoid geometrical errors on the LE, which will be shown later to be crucial for this case.

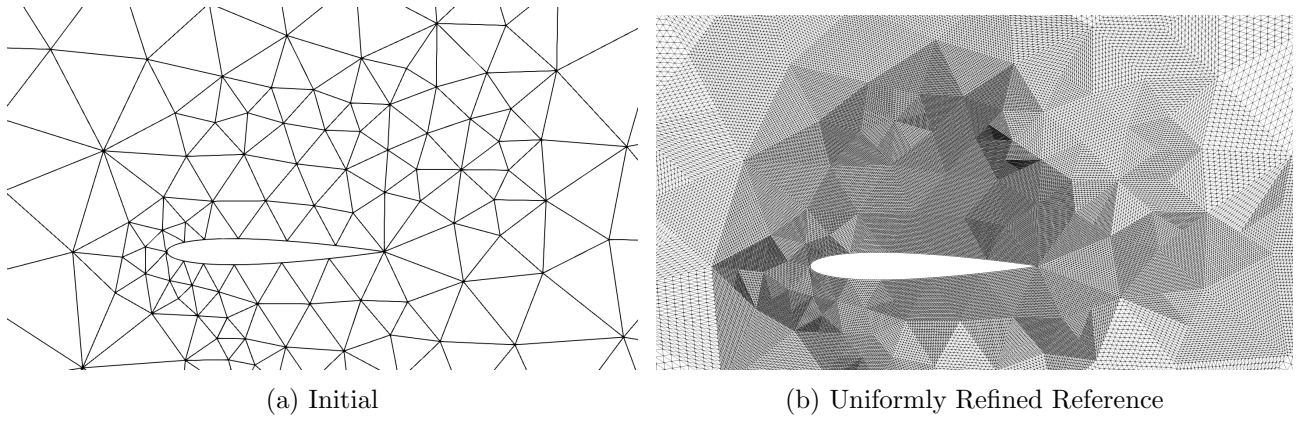
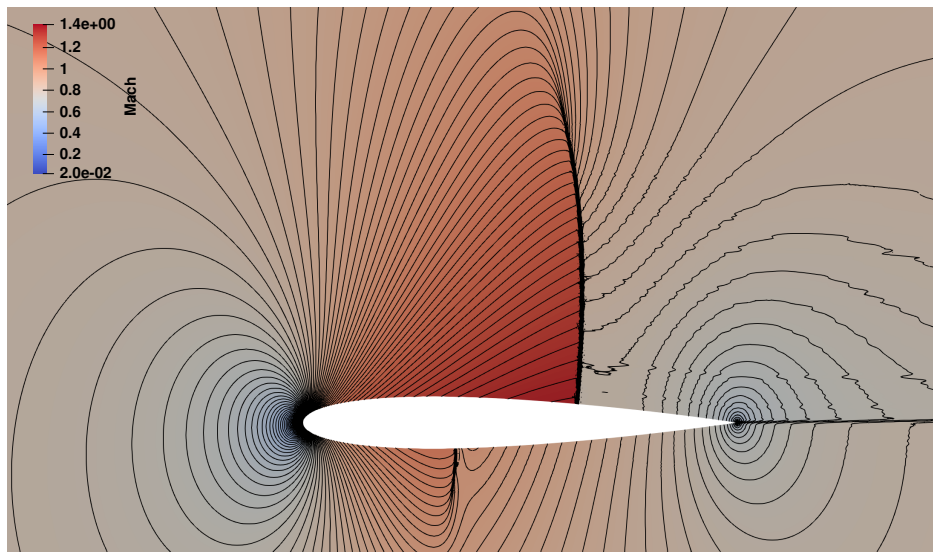
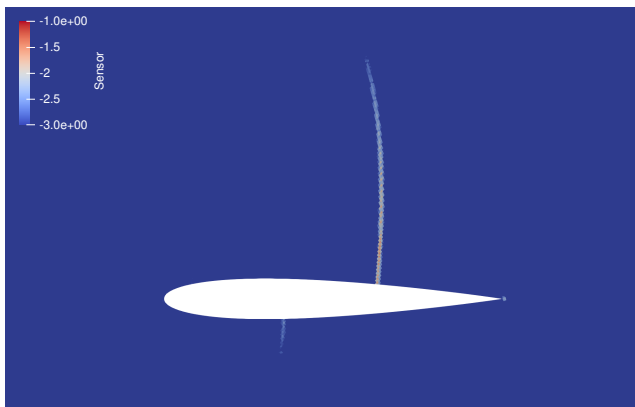


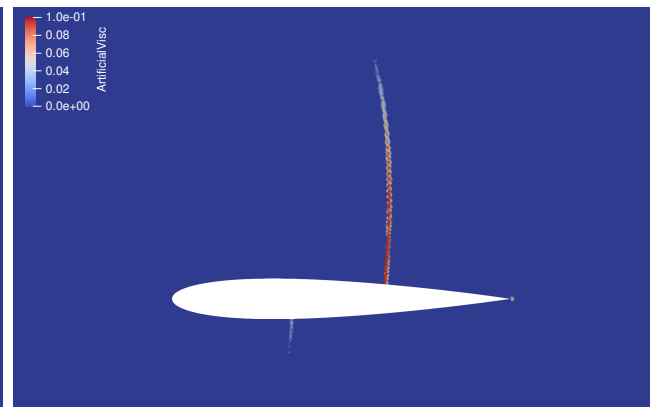
Figure 10.8: Initial mesh with $N_{el}=858$ and four times uniformly refined reference mesh with $N_{el}=318 \times 10^3$



(a) Reference Solution (M)



(b) PP-sensor



(c) Artificial viscosity

Figure 10.9: Reference solution, Mach number with overlaid contours showing continuous flow without artificial boundary layer created on the slip boundary or artificial dissipation added on the LE.

Figure 10.9 a) shows the Mach contours are smooth both around the airfoil and in the far field. They converge to the two shocks and show a minimal oscillatory behaviour downstream of the main shock on the suction side, showing the minimal dissipation we have chosen on the shock will help drive the AMR with the PP-sensor. The pressure side exhibits a weaker shock, the streamlines are smooth and both the PP-sensor and the artificial viscosity are triggered. Therefore, we conclude that such a solution is a good compromise between stability, accuracy and the visibility of sharp shock, something crucial for the AMR. Note that the solver did not apply artificial dissipation on the LE.

10.6.2 h-adaptation

Following the best practices from the intake, we start with the pure h-adaptation as the most flexible approach to capture the shocks. For all simulations in the pure h-adaptation, we start with $p=2$ as this is by far the fastest to reach the steady state and the previous best practice. What is important to note in this case, however, is that due to the curved surface, this automatically means some geometrical discontinuity is created on the leading edge, which as a consequence, will be targeted by the adaptation throughout all AMR iterations. Starting from h_1 , we restrict the h_{min} to create a more uniform mesh around the body of interest, as this is the cheapest step. Then with h_2 the weaker shocks are starting to become clearly distinguishable on the PP-sensor. In both h_1 and h_2 steps, there is a clear overrefinement around the LE of the airfoil, which has been reported in both the straight-sided meshes but also in the high-order ones [5][64][7]. After the h_3 iterations onwards, the h-adaptation is already working mainly on improving the shocks resolution, which can be assumed to be in their asymptotic region. This leads to a final mesh size of h_5 with $N_{el} = 4855$, which is only 1.5% of the reference one, but with the same resolution on the shocks.

Finally, for the last two solutions (h_4 and h_5) we increase the artificial viscosity multiplier μ_0 inside the shock to $\mu_0 = 0.06$ to locate the shocks on their final location and increase the polynomial order uniformly to $p=3$ for the h_5 to match the reference order, reduce the geometrical aliasing on the LE and also avoid the useful for AMR-identification dispersion errors on the shocks. This results in $N_{DoF} = 49 \times 10^3$ and the solver is run for another $\Delta t = 0.1s$. As expected, the standard h-adaptation is capable of resolving both shocks to the resolution of the reference mesh, and the solution is qualitatively similar in terms of the PP-sensor. However, a more careful inspection of the LE of the airfoil shows high PP-Sensor values on the leading edge that is not present in the reference solution.

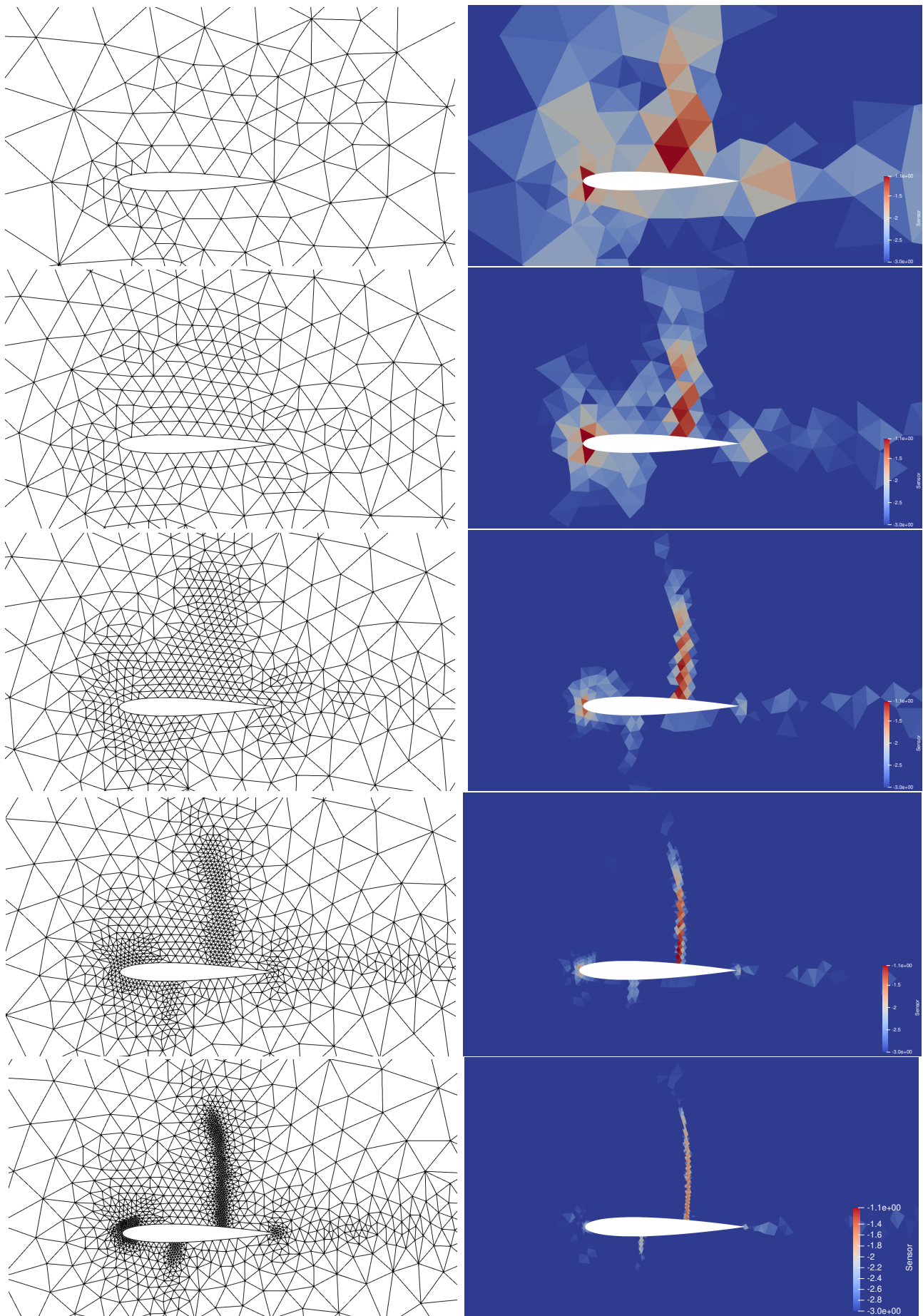


Figure 10.10: h-adaptation - mesh and PP-sensor solution evolution. Please note the indicator is limited to $[-3, -1.1]$ for approximate visualization purposes of what elements are proposed to be split.

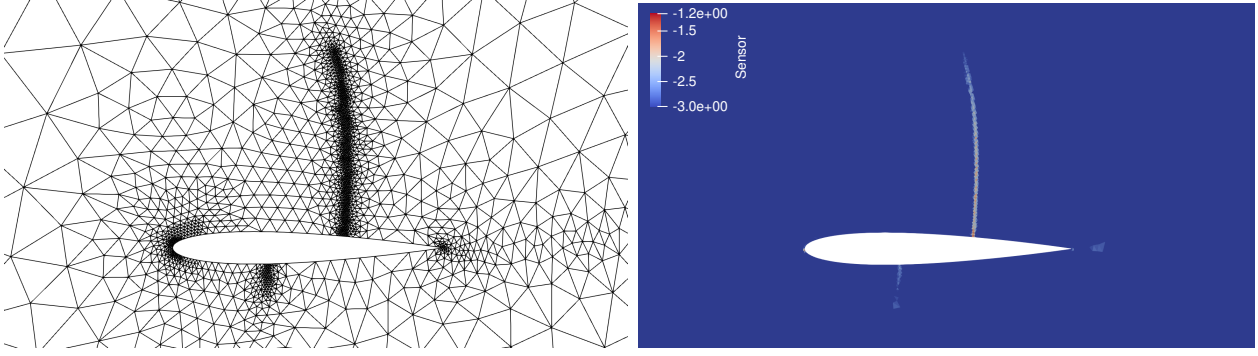


Figure 10.11: Final h-adaptation - mesh and PP-sensor solution evolution. Please note the high PP-sensor on the LE and that the indicator is limited to $[-3, -1.1]$ for consistency.

10.6.3 hp-adaptation

A natural evolution to the uniform $p=3$ increase is to use a p -adaptation at the end of the h_5 . While straightforward and successful in our tests, this misses the opportunity to also solve the LE overrefinement and to further decrease the overall cost of the simulation in terms of elements marked for adaptation. Specifically, the ones further away from the shocks, likely caused by the PP-sensor capturing smooth features. In our approach, we take advantage of the fact that the PP-sensor is inversely sensitive to the changes of the order p . Therefore, we propose a p -adaptation, increasing p by one, after every h -adaptation. Only based on this p -adapted solution, we mark the elements for the next cycle of h -adaptation and apply it. Our experiments show that the PP-values around the discontinuities remain high, as expected by the PP-sensor and the Gibbs phenomenon. The ones in the smooth part of the field that were initially marked for adaptation do show lower PP-values and often go below the adaptation threshold. Moreover, this also drastically decreases the geometrical aliasing that comes with $p=2$, leading to a better captured stagnation point region around the LE earlier in the adaptation cycle. All this, while keeping the high Δt for the initialization transient after the h -adaptation. While adding an extra simulation time for the p -cycle, this was found to be highly efficient, especially in the coarse refinement steps, helping us maintain low DoF throughout the AMR.

Such an area that benefits from the hp -adaptation is the acceleration region on the pressure side of the airfoil. As this is a smooth feature, it does allow for the p -refinement to resolve it sufficiently even at h_3 . There is a potential for the weaker shock to suffer from underresolution if sensor values drop under the PP-threshold after the p -refinement cycle, but the use of minimal artificial dissipation during the AMR process helps keep the elements adapted and achieves comparable resolution to the h . Last, the LE in this case does not suffer from high PP-sensor values. In our hp -adaptation test-case, this process resulted in an additional 25% reduction in terms of element count $N_{el}=3667$ and $N_{DoF}=31 \times 10^3$ or 37% DoF reduction compared to the pure h -adaptation.

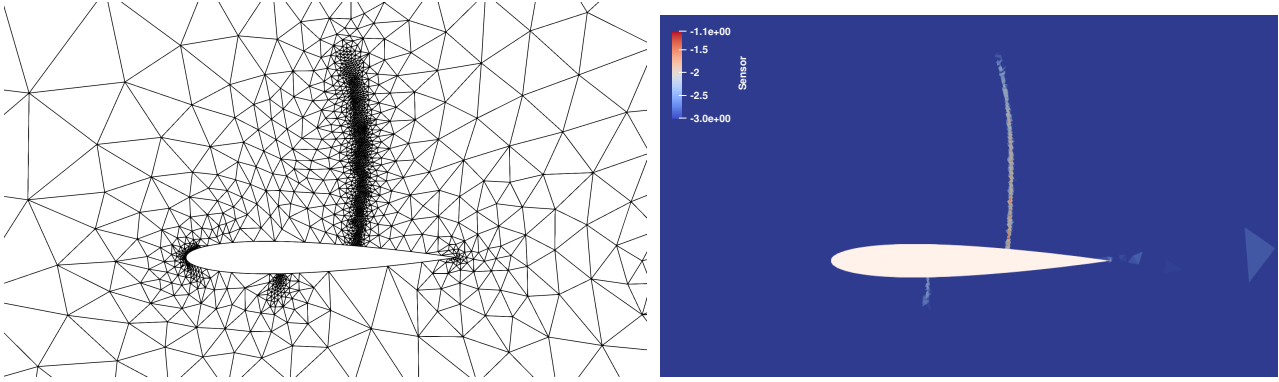


Figure 10.12: hp-adaptation - mesh and PP-sensor solution. Please note the indicator is limited to $[-3, -1.1]$ for consistency.

Given the computational cost (DoF) savings and the capability to maintain both the mesh resolution in the shocks, decrease the geometrical errors and resolve smooth features of the simulation, we recommend adapting this p-adaptation (p-cycling) after every h-adaptation, especially for the initial cheaper AMR-iterations, where the cost of the additional p-adaptation is minimal. This way creating a more efficient hp-adaptation.

10.6.4 hrp-adaptation

Finally, at this point, we have analysed all combinations necessary to assemble a full hrp-AMR that benefits from all mesh modifications and the variable polynomial order. This combination starts with the hp-adaptation with p-cycling for resolving the shocks, while clearing out the smooth features and geometrical aliasing. Last, rp-AMR is applied after the hp, aiming at further increasing accuracy without adding to the DoF computational cost. We use the same stopping criteria for the transition from hp to rp-adaptation as for the intake, a function of h_{min} , two times the desired h_{min} .

Having this proof-of-concept methodology, we end up with the same hp-adaptation until the fourth AMR step - hp_4 , followed by two cycles of rp-adaptation. This is enhanced by the mesh sliding on the CAD, where the link is obtained by CAD-reconstruction, which further demonstrates the synergies between our AMR approach and Part I. The final mesh has a similar resolution to the pure h-adaptation and the reference around shock regions, especially around the surface. Hence, the proposed h-r-p adaptation does manage to maintain the benefits of the previous AMR configurations, such as avoiding overrefinement in the acceleration region and downstream the weaker shock, which inherits from the hp. Last, to ensure sufficient resolution on the leading edge, we split the two elements connected on the stagnation vertex on the LE. The mesh and the PP-sensor after the final p-adaptation can be seen in Figure 10.13. The process leads to only $N_{el}=2155$ and overall $N_{DoF} = 18 \times 10^3$. This is a further 41% reduction in terms of elements and 41% reduction in terms of DoF compared to the hp. Last, compared to the uniformly refined reference solution, this is a 148 times reduction in element count and a 175 times reduction in terms of DoF.

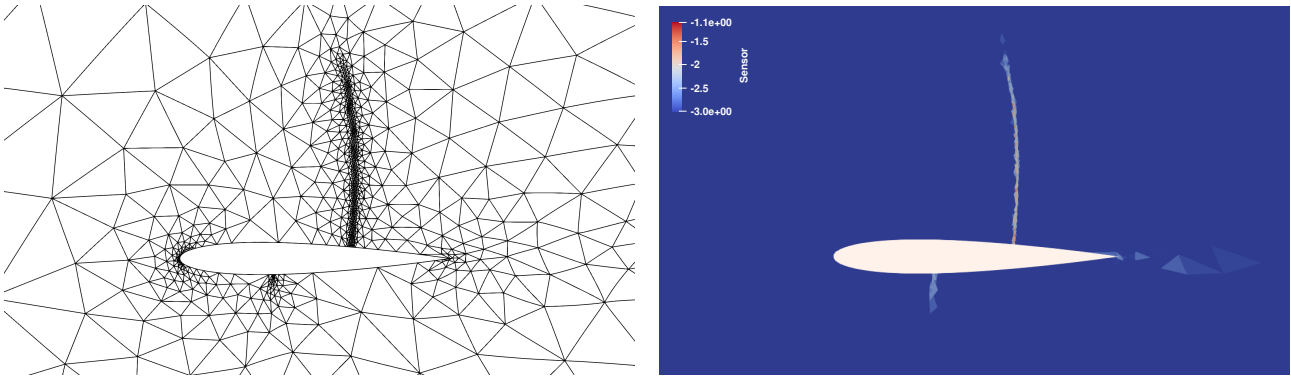


Figure 10.13: hrp-adaptation - mesh and PP-sensor solution. Please note the indicator is limited to $[-3, -1.1]$ for consistency.

What must be acknowledged, is the tendency for overrefinement on the LE, where elements become smaller than those in the main shock and could become the CFL-limiting region for the simulation, especially if r-adaptation is applied even further. More importantly for the discussion, the PP-sensor does resemble the hp-solution - there are no high PP-values around the LE, but there is a wider region marked around the weak shock inside the domain. While qualitatively acceptable, this raises the question of the accuracy of the shock capturing, which we will address further.

10.6.5 Solution Accuracy

To scrutinize the AMR-solutions further, we analyze the accuracy by comparing all presented configurations, both on the airfoil wall and the flow inside the domain.

It is important to examine not just the flow characteristics on the wall, but also the flow inside the domain. Therefore, we compare the Mach contours between the final solutions, see Figure 10.14. All three configurations show a similar pattern as the reference case - continuous streamlines, starting from the LE and converging inside the shocks. Some minor noise is present immediately downstream of the main shock, but smooth contours downstream of the weak one.

Recalling the discussion on the high PP-value present on the LE of the h-adaptation, this translates to adding artificial dissipation on the LE and the artificial deceleration around the airfoil, starting from the LE. This can be observed clearly on the h-adaptation M streamlines and has also been reported by Balan and May [5] [64] .

Contrary to the h-adaptation, no artificial boundary layer is created in the hp and hrp configurations, suggesting an accurate geometrical representation, a good overall use of our artificial viscosity and hence no excessive dissipation on the LE. An important observation on these figures is the smooth streamlines in the acceleration region and in the far field in both the hp and hrp-adaptations. Despite being extremely underresolved in terms of h-, the p-refinement in these regions is sufficient to create a continuous solution at a fraction of the cost, supporting the hypothesis that the additional p-adaptation cycling is preferable for smooth features.

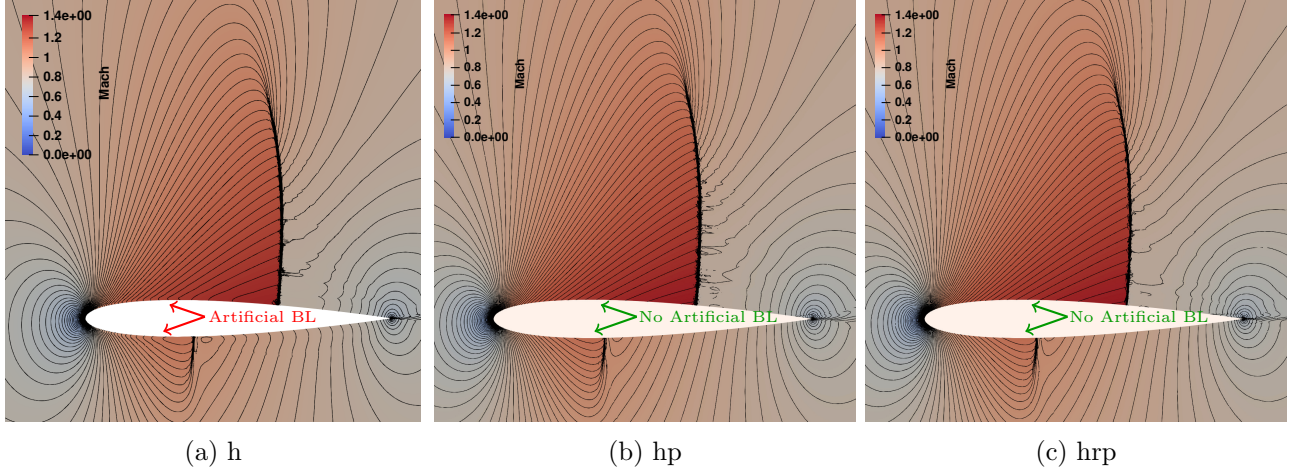


Figure 10.14: Comparison between the different AMR final solutions showing smooth streamlines and sufficient resolution inside the domain. Please note the artificial deceleration around the airfoil for the h-adaptation, starting from the LE. The hp and hrp, on the other hand, show straight Mach profiles starting from the body, as expected for the slip boundary condition.

A careful analysis of the Mach number on the airfoil can reveal an accurate picture of the boundary resolution, the shock capturing, and for any discontinuities in the derivative of the solution on the curved geometry, see Figure 10.15. Starting from the LE and the acceleration regions, all four cases begin with zero velocity at the stagnation point and the flow accelerates until it reaches the shocks. Both the hr and hrp-AMR have almost a complete overlap with our reference solution. The h-adaptation is the only outlier with a slight, but noticeable, $\Delta M < 0.01$ drop compared to the reference case.

All AMR-configurations capture the main shock on the suction side of the airfoil accurately without major over- or undershoots with a sharp profile. Compared with the reference solution, we observe a minor offset $\frac{\Delta x}{c} < 2.5 \times 10^{-3}$ for the h- and hp-AMRs and a near complete overlap between the hrp-AMR and the reference solution.

On the pressure side, where the weaker shock is harder to capture with AMR, all of our AMR-configurations have resolved it. In more details, the h-adaptation Mach number drops both before the shock, but also downstream in the recovery region. This is consistent with the streamlines, as the additional artificial dissipation on the LE propagates downstream and hence indicates the sensitivity of the LE. The hp-AMR does improve the agreement with the reference before and after the shock and exhibits the expected sharp shock. Nevertheless, the hp-AMR does move it $\frac{\Delta x}{c} < 5 \times 10^{-3}$ upstream, showing a potential for further improvements. Lastly, the hrp-AMR has the best agreement against the uniform reference solution, accurate shock location and identical Mach profiles upstream the shock and in the recovery region.

All in all, despite the minor discrepancies, all three AMR-configurations show an overall good agreement with the reference, both from their shock capturing, streamlines, but also integral values: a maximum drag deviation for all cases - the h-adaptation $\Delta C_d = 2\%$ and the best agreement - for the hrp-adaptation $\Delta C_d = 0.4\%$, all these at a fraction of the reference DoF.

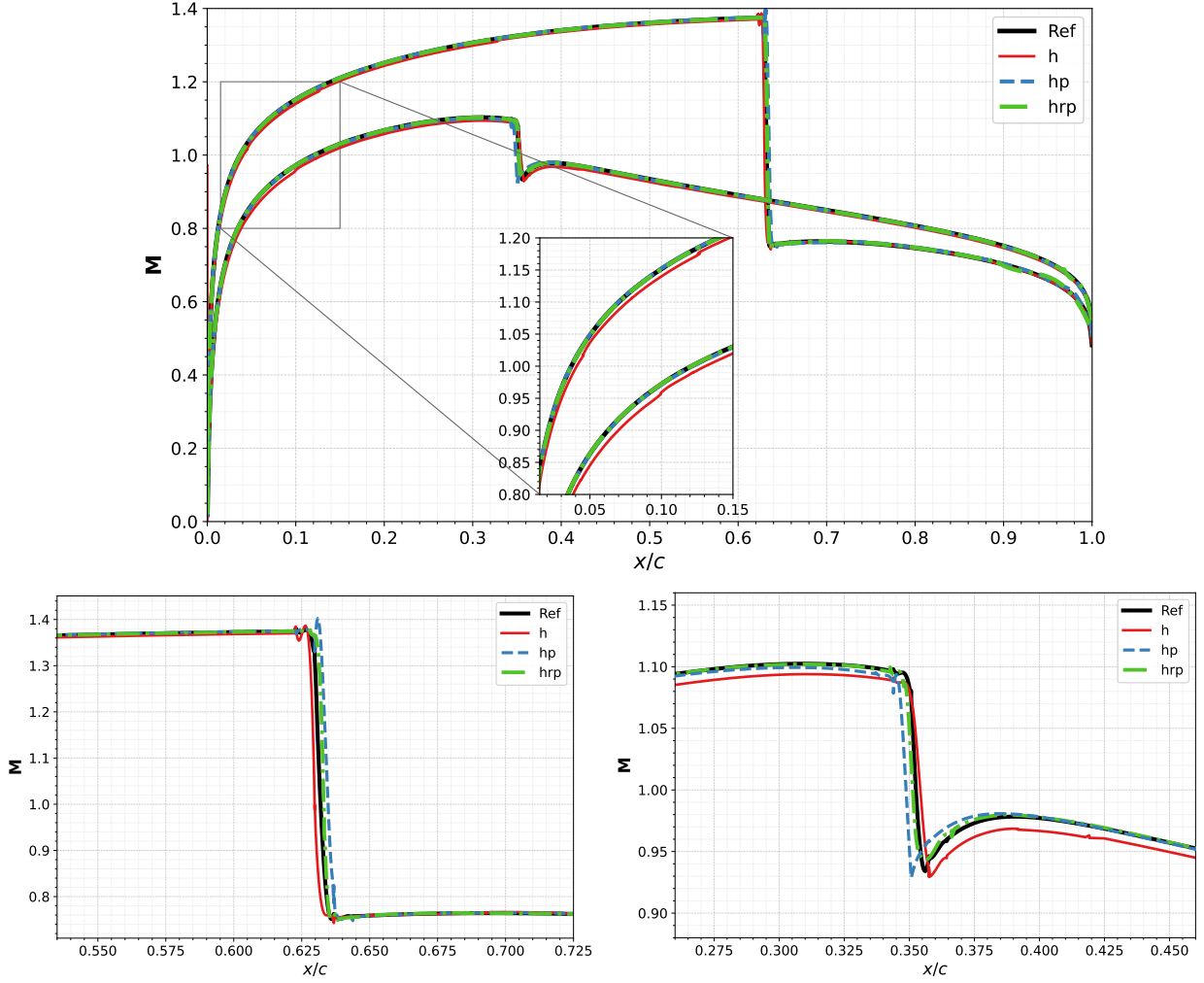


Figure 10.15: Comparison of the reference solution with the h, hp and hrp Mach number on the airfoil. Please note the almost full overlap between the hrp and the reference solution and the effect of LE artificial dissipation for the h-adaptation, offsetting both the acceleration region and the recovery region downstream the weaker shock.

In conclusion, this example demonstrated the benefits of combining the hr-adaptation with p-cycling on a multistep adaptation process, a phenomenon not widely reported in the literature. The proposed p-cycling after every h-adaptation significantly decreased the overreliance on the less efficient hr-adaptation in the smooth regions. Moreover, even when starting from very coarse meshes, the framework was capable of converging close to the reference solution without an artificial boundary layer forming due to LE artificial dissipation, also due to better capture of the geometry. All this supports the hypothesis that the hrp-configuration can be the preferred choice for achieving an optimal hp-AMR, given the trade-off between accuracy and cost.

Last, the author acknowledges that this was a proof-of-concept and a full production ready framework will likely result in an adaptation driver that estimates an optimal hp-resolution and even bypasses some of the hp-cycling steps, especially after hp_3 and integrates it fully with the solver. All these activities suggest further potential for DoF gains and also a potential for time-to-solution performance improvements. This analysis, however, is outside of the scope of this thesis and is left for future work.

Chapter 11

Conclusions and Future Work

11.1 Conclusions

In Part II, we have targeted three main goals: create a generic, scalable, high-order curvilinear h-adaptation framework. To investigate if this framework can be used to converge the extremely coarse meshes from Part I to the correct physical solution and to provide a proof-of-concept for an hrp-adaptation, where all high-order mesh modification techniques of NekMesh are combined into a single framework for a-posteriori AMR.

High-Order Isoparametric h-adaptation We have presented a new implementation for high-order conformal h-adaptation in both 2D and 3D that follows the ideas of Lohner with bisection/splitting leveraging predefined maps. However, in contrast to the many legacy implementations for straight-sided elements, we extend this to curvilinear ones where the splitting is performed in the reference element with the maximum degree of locality by using predefined reference maps. Through some canonical examples in both 2D and 3D we showed that our implementation has several important properties: preserves geometrical accuracy, mesh validity and provides good performance scaling between polynomial orders, allowing us to avoid expensive iterative CAD operations near the curved surface. Then, looking from a mesh quality perspective, we demonstrated that both in 2D and 3D an isotropic 1:4 or 1:8 maps for distorted triangle or tetrahedral, respectively, do improve the mesh quality (J_{sc}). This can be exploited during the AMR process, supporting the hypothesis of Part I for very coarse initial meshes. To ensure conformity in the process, a propagation algorithm is used during the edge splitting, which in 3D allows for bypassing the need for all 240 reference maps. Last, to improve the mesh quality in terms of vertex valences, high-order swapping and smoothing are used in 2D and smoothing in 3D.

Tackling the Curse of Asymptotic Refinement In this work, we demonstrated that it is possible to start from a very coarse mesh that does not resolve the flow features of interest and combining isotropic h-refinement with a feature-based sensor, one can reach the asymptotic region from a very coarse mesh. This was sufficient for capturing the Laminar Separation Bubble in the incompressible flow around the extruded IFW, but also in the compressible case for the transonic NACA0012, where starting from a mesh with only 12 boundary edges around the airfoil, it managed to resolve adequately the shocks after the adaptation.

The h-adaptation of the extruded IFW led to 21% reduction in the number of DoF used in the simulation compared to a CFD expert remeshing. In 2D, we started our adaptation verification with h and hr-adaptation on a supersonic intake, where both the pure h-adaptation and the hr-AMR are capable of resolving shocks of different strengths located in a very confined region with the hr-AMR being 46% cheaper than the pure h-adaptation and 19 times cheaper than the uniform refinement. Of course, this does not translate to the same time-to-solution as five cheaper meshes are simulated, with the time-to-solution being a fraction of the uniform case. This supports our initial hypothesis that starting from extremely coarse curved meshes with good geometrical accuracy is sufficient to reach the correct physical results in a cheaper and automatic manner.

Towards hp/SEM for compressible flows Finally, we have attempted to create a framework that employs all benefits of the high-order methods through the combination of the h,r, and p-adaptation. By exploring two inviscid steady compressible cases, we iterated through h/p-adaptation configurations, all starting from very coarse meshes for the NACA0012 transonic case and the supersonic intake. From there, we can provide both high-level and low-level best practices for anyone attempting hp-adaptation.

For these best practices, we will consider three different stages in the simulation - first, where the initial features are indistinguishable and where the shock discontinuities and other flow features should be located and resolved. A second one, where the shocks will already be clearly identified in narrow regions, but the maximum spatial discretization is not yet achieved. A final one, where the shocks are fully resolved, and the smooth features are of interest. For the first one, we found the flexibility of multilevel h-adaptation while running the lowest possible order to reach the asymptotic region most robustly. Then, only when the asymptotic region is reached, where the features are clearly defined and the sensors mark a narrow band of elements for refinement (two times wider than Δh_{min}), we recommend proceeding with the r-adaptation. This way, one can gain the benefits of the improved resolution and often align the mesh to an anisotropic one around the shock/discontinuity without adding extra DoF. The last stage, when the shocks and features are correctly located and resolved, is p-adaptation for the smooth features.

A final enhancement of the process is by applying a single p-refinement after every h-step. This was found to be beneficial for resolving smooth features earlier in the adaptation and resulting in nearly 1.6 times reduction in DoF compared to the standard h-adaptation.

When all techniques are combined together in the proposed hrp-adaptation, we achieved an overall reduction of 175 times in terms of DoF compared to a uniform refinement for the steady NACA0012. Analyzing the solution accuracy, the proposed AMR converged to a physical solution without an artificial boundary layer forming due to LE artificial dissipation, a common issue for this case, resolved both shocks and achieved a good mesh resolution there. While not an optimal hrp-AMR process, all this supports the initial hypothesis that the combination of all mesh modification techniques with a variable polynomial order into a true hrp-AMR results in a more efficient utilization of high-order methods for compressible flows.

11.2 Current & Future Work

Although beyond the scope of this thesis, several activities are ongoing to extend the proposed methodology.

First, the author aims to provide (y^+) control for AMR by adding additional transitional prism maps, as stated in Part I. This will also allow regions with variable numbers of prism layers. As more than 50% of the elements are prisms in most industrial cases, this will result in a significant efficiency gain.

Second, a high-order swapping operator in 3D that preserves the mesh validity and the element curvature is of great interest in allowing more than three levels of refinement in a narrow region, such as shocks or discontinuities. Third, a mesh coarsening through implicit hashing of the parent-child element relationship is expected to allow for a more precise interpolation and possibly dynamic mesh adaptation with mesh coarsening up to the base mesh.

Last, we point out the current activities in the group for moving the NekMesh data structure to the Nektar++ one, which will allow for a direct MPI parallelism, direct integration to the solver utilities for dynamic AMR with conservative interpolation facilitated by Nektar++ utilities during solve time.

From AMR incompressible applications, while we presented 3D AMR for capturing the transition, there are already activities for 3D h, hr and hp-adaptation for improving the vortical structures in cases such as NACA0012 Wingtip by J.Zhou. The initial results are promising, and once mature, we plan to proceed to the full IFW *a posteriori* AMR for capturing both transition and the vortical structures. This, however, also requires a robust feature-based indicator, a Spectral Error Indicator such as the one by Mavriplis [63] or even an adjoint one, as the cost of the case will be prohibitive to iterate through numerous AMR cycles.

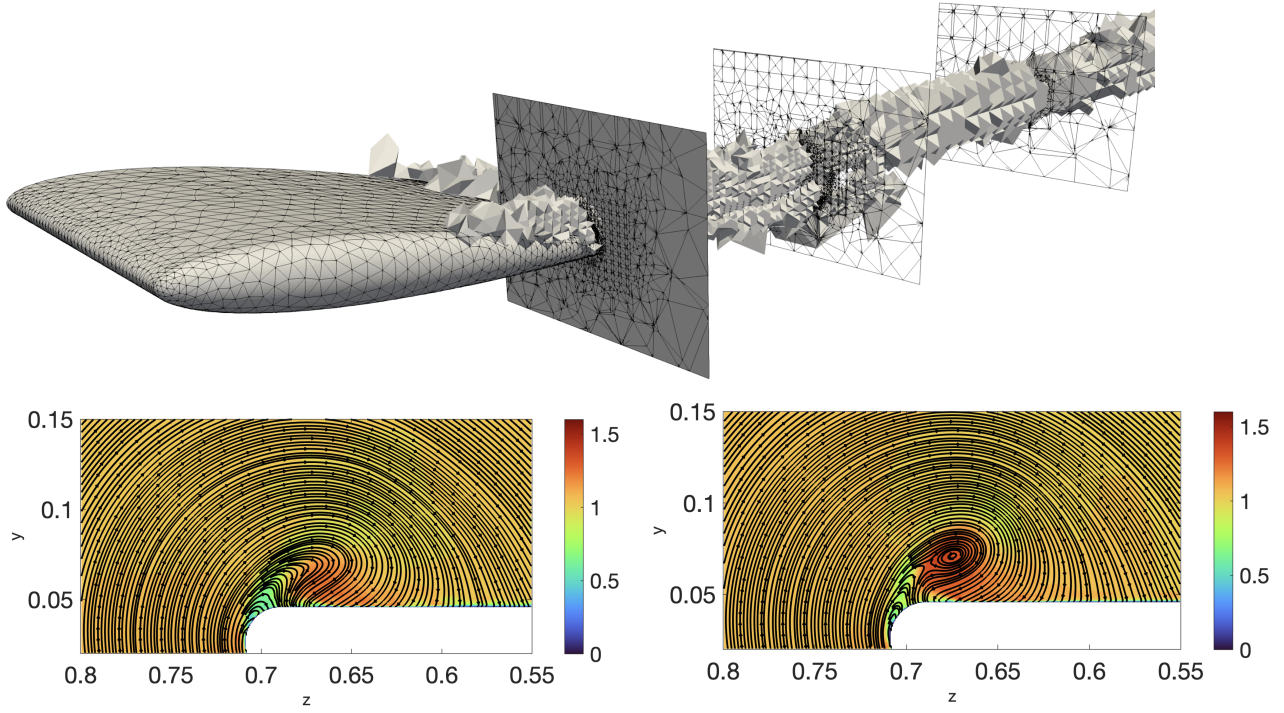


Figure 11.1: h-adaptation on NACA0012 wingtip on vortical dominated flow driven by Q-criterion and the improvement in vortical resolution on the wing tip trailing edge. [46]

Finally, for compressible flow cases, we plan to extend the h-r-p methodology for turbomachinery applications for the full compressible N-S with transonic buffet. In this case, the shockwave boundary layer interactions will make the shockwave unsteady and hence require dynamic h-r-p AMR. To achieve that, we will need the full integration of the Nektar++ - NekMesh, local conservative interpolation, but also a better performing r-adaptation. Moreover, an anisotropic r-adaptation driven by optimal mesh metric, $f(\epsilon)$, will likely provide further performance gains compared to the heuristic approach in our h-r-p adaptation. Last, the author believes that the choice for the “best” combination of h-r-p adaptation and the target mesh metric could be reformulated as message passing graphics neural network that will be faster, more robust and automatic, bypassing the iterative threshold tuning and the expensive metric calculation, while still achieving near-optimal efficiency of the final hp-resolution.

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] V. Anbumani, S.E. James, and P.R. Eiseman. High Lift CRM locally structured grid generation with GridPro. In *2018 AIAA Aerospace Sciences Meeting*, 01 2018.
- [3] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cervený, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, W. Pazner, M. Stowell, V. Tomov, I. Akkerman, J. Dahm, D. Medina, and S. Zampini. Mfem: A modular finite element methods library. *Computers and Mathematics with Applications*, 81:42–74, 2021. Development and Application of Open-source Software for Problems with Numerical PDEs.
- [4] A. Balan, M.A. Park, S.L. Wood, W.K. Anderson, A. Rangarajan, D.P. Sanjaya, and G. May. A review and comparison of error estimators for anisotropic mesh adaptation for flow simulations. *Computers & Fluids*, 234, 2 2022.
- [5] A. Balan, M. Woopen, and G. May. Hp-adaptivity on anisotropic meshes for hybridized discontinuous galerkin scheme. In *22nd AIAA Computational Fluid Dynamics Conference*, June 2015.
- [6] F. Basile, J-B. Chapelier, R. Laraufie, and P. Frey. Adaptive hybrid rans/les simulations for unstructured meshes with the discontinuous galerkin method. In *AIAA Scitech 2022 Forum*, 01 2022.
- [7] 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.
- [8] Beta CAE Systems. ANSA v23.1.0 User Guide and Version Release Webinar, 2025. High-order meshing capabilities.
- [9] L. Branets and G. Carey. Condition number bounds and mesh quality. *Numerical Linear Algebra with Applications*, 17(5):855–869, 09 2010.

- [10] C. J. Budd and R. E. Walsh. The geometry of r-adaptive meshes generated using optimal transport methods. *Journal of Computational Physics*, 282:113–137, 2015.
- [11] F.F. Buscariolo, J. Hoessler, D.Moxey, A. Jassim, K. Gouder, J. Basley, Y. Murai, G.R.S. Assi, and S.J. Sherwin. Spectral/hp element simulation of flow past a formula one front wing: Validation against experiments. *Journal of wind engineering and industrial aerodynamics*, 221:104832, 02 2022.
- [12] 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.
- [13] Capgemini Engineering. Opencascade technology. <https://www.opencascade.com>, 2023. Software platform for 3D CAD, CAM, CAE. Last accessed: January 2023.
- [14] J. Cervenu, V. Dobrev, and T. Kolev. Nonconforming mesh refinement for high-order finite elements. *SIAM Journal on Scientific Computing*, 41:367–392, 2019.
- [15] W. Chen, S. Sun, Y. Wang, N. Lei, C. Sadasivan, A. Tassiopoulos, S. Chen, H. Si, and X. Gu. Robust surface remeshing based on conformal welding. In *Proceedings of the 2024 International Meshing Roundtable (IMR)*, 2024.
- [16] S. Chun, J. Marcon, J. Peiró, and S.J. Sherwin. Reducing errors caused by geometrical inaccuracy to solve partial differential equations with moving frames on curvilinear domain. *Computer Methods in Applied Mechanics and Engineering*, 398:115261, 2022.
- [17] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378, 2014.
- [18] E. de Oliveira Carvalho, A.F. de Castro da Silva, and R.C. Moura. A performance analysis of multiple feature-based indicators for adaptive mesh refinement in continuous galerkin simulations. In *AIAA Scitech 2025 Forum*, 2025.
- [19] Department of Aeronautics of the Imperial College London, Scientific Computing and Imaging Institute of the University of Utah. *Nektar++: Spectral/hp Element Framework - User Guide*, May 2024.
- [20] V. Dobrev, P. Knupp, T. Kolev, K. Mittal, R. Rieben, and V. Tomov. Simulation-driven optimization of high-order meshes in ale hydrodynamics. *Computer & Fluids*, 208:104602, 08 2020.
- [21] V. Dobrev, P. Knupp, T. Kolev, K. Mittal, and V. Tomov. The target-matrix optimization paradigm for high-order meshes. *SIAM Journal on Scientific Computing*, 41:50–68, 2019.

- [22] V. Dobrev, P. Knupp, T. Kolev, K. Mittal, and V. Tomov. hr-adaptivity for nonconforming high-order meshes with the target matrix optimization paradigm. *Engineering with Computers*, 38(4):3721–3737, 08 2022.
- [23] V. Dolejsi and G. May. *Anisotropic hp-Mesh Adaptation Methods, Theory, implementation and applications*. Birkhäuser Cham, 01 2022.
- [24] T. Dzanic, T. Kolev, and K. Mittal. A method for bounding high-order finite element functions: Applications to mesh validity and bounds-preserving limiters. *Journal of Computational Physics*, 2026.
- [25] T. Dzanic, K. Mittal, D. Kim, J. Yang, S. Petrides, B. Keith, and R. Anderson. Dynamo: Multi-agent reinforcement learning for dynamic anticipatory mesh optimization with applications to hyperbolic conservation laws. *Journal of Computational Physics*, 506:112924, 2024.
- [26] D. Ekelschot. Mesh adaptation strategies for compressible flows using a high-order spectral/hp element discretisation. *PhD Thesis, Spiral - Imperial College London*, 2016.
- [27] 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 and Structures*, 181:55–69, 2017.
- [28] OpenFOAM ESI. OpenFOAM, API guide: `ccmToFoam.C` file reference. Last accessed October 2023.
- [29] P. Farrell, M. D. Piggott, C. Pain, G. Gorman, and C. Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, 198(33–36):2632–2642, 2009.
- [30] E. Ferrer, G. Rubio, G. Ntoukas, W. Laskowski, O.A. Mariño, S. Colombo, A. Mateo-Gabín, H. Marbona, F. Manrique de Lara, D. Huergo, J. Manzanero, A.M. Rueda-Ramírez, D.A. Kopriva, and E. Valero. A high-order discontinuous galerkin solver for flow simulations and multi-physics applications. *Computer Physics Communications*, 287:108700, 2023.
- [31] K.J. Fidkowski and D.L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49:673–694, 2011.
- [32] A. Gargallo-Peiró, X. Roca, and J. Sarrate. Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 103:342–363, 2015.

- [33] T. Gerhold and J. Neumann. The parallel mesh deformation of the DLR TAU-code. In C. Tropea et al., editor, *New Results in Numerical and Experimental Fluid Mechanics VI*, pages 162–169. Springer Berlin Heidelberg, 2008.
- [34] C. Geuzaine and J.F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 09 2009.
- [35] M. D. Green, K. S. Kirilov, M. Turner, J. Marcon, J. Eichstäedt, E. Laughton, C. D. Cantwell, S. J. Sherwin, J. Peiro, and D. Moxey. NekMesh: An open-source high-order mesh generation framework. *Computer Physics Communications*, 298, 2024.
- [36] J. Holke, J. Markert, D. Knapp, L. Dreyer, S. Elswijker, N. Böing, I. Lilikakis, J. Fussbroich, T. Leistikow, F. Becker, V. Uenlue, O. Albers, C. Burstedde, A. Basermann, C. Hergl, J. Weber, K. Schoenlein, J. Ackerschott, E. Andreev, Z. Csati, A. Dutka, B. Geihe, P. Kestener, A. Kirby, H. Ranocha, and M. Schlottke-Lakemper. t8code - modular adaptive mesh refinement in the exascale era. *Journal of Open Source Software*, 10(106):6887, 2025.
- [37] D. Huergo, G. Rubio, and E. Ferrer. A reinforcement learning strategy for p-adaptation in high order solvers. *Results in Engineering*, 21:101693, 2024.
- [38] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.
- [39] ITI-Global. CADfix: CAD translation, healing, repair, and transformation. <https://www.iti-global.com/cadfix>, 2017. Software tool for CAD model translation, repair, and optimization. Last accessed: January 2023.
- [40] 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, 04 2006.
- [41] S. Joshi, J. Kou, A. Hurtado de Mendoza, K. Puri, C. Hirsch, G. Rubio, and E. Ferrer. Length-scales for efficient CFL conditions in high-order methods with distorted meshes: Application to local-timestepping for p-multigrid. *Computers & Fluids*, 265:106011, 2023.
- [42] G Karniadakis and S J Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
- [43] G. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2nd edition, 06 2005.

- [44] P. Khanteimouri and M. Campen. 3D Bezier Guarding: Boundary-conforming curved tetrahedral meshing. *ACM Trans. Graph.*, 42(6), 12 2023.
- [45] P. Khurana, S. Sherwin, J. Hoessler, D. Moxey, and A. Chatzopoulos. Lower-order re-fined preconditioning for spectral/hp element methods for complex, 3D geometries. In *Proceedings of the 9th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2024)*.
- [46] K.S. Kirilov, J. Zhou, and T. Stokes. Meshing and mesh adaptation session. *Nektar++ workshop 2025*, 10.52843/cassyni.715s01, 2025.
- [47] P. Knupp. Introducing the target-matrix paradigm for mesh optimization via node-movement. *Eng. with Comput.*, 28(4):419–429, 10 2012.
- [48] P. M. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3):217–241, 2003.
- [49] M. Kompenhans, G. Rubio, E. Ferrer, and E. Valero. Comparisons of p-adaptation strategies based on truncation and discretization errors for high-order Discontinuous Galerkin methods. *Computers & Fluids*, 139:36–46, 11 2016.
- [50] D.A. Kopriva, F.J. Hindenlang, T. Bolemann, and G.J. Gassner. Free-stream preservation for curved geometrically non-conforming discontinuous galerkin spectral elements. *Journal of Scientific Computing*, 79(3):1389–1408, 06 2019.
- [51] E. Laughton, G. Tabor, and D. Moxey. A comparison of interpolation techniques for non-conformal high-order discontinuous galerkin methods. *Computer Methods in Applied Mechanics and Engineering*, 381:113820, 8 2021.
- [52] A. I. Liosi, P. Khurana, S. J. Sherwin, J. Hoessler, A. Swift, A. Chatzopoulos, and F. Bottone. Requirements & current capabilities for implicit large eddy simulation of the imperial front wing using spectral h/p element methods. *Proceedings of DLES14*. Springer Cham, (Accepted) 2025.
- [53] A.I. Liosi, H. Wustenberg, P. Khurana, S.J. Sherwin, J. Hoessler, A. Swift, and A. Chatzopoulos. Comparing efficient implicit time-stepping techniques for highly resolved simulations using industrial geometries. In *AIAA Scitech 2025 Forum*.
- [54] C. Lock, O. Hassan, R. Sevilla, and J. Jones. Predicting the near-optimal mesh spacing for a simulation using machine learning. In *SIAM International Meshing Roundtable 2023*, pages 115–136, Cham, 2024. Springer Nature Switzerland.
- [55] A. Loseille and F. Alauzet. Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, Jan 2011.

- [56] A. Loseille and F. Alauzet. Continuous Mesh Framework Part II: Validations and Applications. *SIAM Journal on Numerical Analysis*, 49(1):61–86, Jan 2011.
- [57] A. Loseille, F. Alauzet, and V. Menier. Unique cavity-based operator and hierarchical domain partitioning for fast parallel generation of anisotropic meshes. *Computer-Aided Design*, 85:53–67, 2017.
- [58] A. Loseille and R. Löhner. Cavity-based operators for mesh adaptation. *Proceedings of 51st AIAA Aerospace Sciences Meeting*, pages 1–8, 2013.
- [59] R. Löhner. Applied CFD Techniques- An Introduction based on Finite Element Methods. *John Wiley and Sons*, March 2008.
- [60] J. Marcon. Mesh adaptation for high-order flow simulations. *PhD Thesis, Spiral - Imperial College London*, page 145, 2020.
- [61] J. Marcon, G. Castiglioni, D. Moxey, S.J. Sherwin, and J. Peiró. rp-adaptation for compressible flows. *International Journal of Numerical Methods in Engineering*, 121:5405–5425, 2020.
- [62] J. Marcon, M. Turner, J. Peiró, D. Moxey, C. R. Pollard, H. Bucklow, and M. Gammon. High-order curvilinear hybrid mesh generation for CFD simulations. In *2018 AIAA Aerospace Sciences Meeting*, 2018.
- [63] C. Mavriplis. Adaptive mesh strategies for the spectral element method. *Computer Methods in Applied Mechanics and Engineering*, 116(1):77–86, 1994.
- [64] G. May and A. Jameson. A kinetic scheme for the navier-stokes equations and high-order methods for hyperbolic conservation laws. *Doctoral Thesis Stanford University*, 01 2006.
- [65] G. Mengaldo, D. Moxey, M. Turner, R. C. Moura, A. Jassim, M. Taylor, J. Peiró, and S. J. Sherwin. Industry-relevant implicit large-eddy simulation of a high-performance road car via spectral/ hp element methods. *SIAM Review*, 63:723–755, 2021.
- [66] A. Milli and S. Shahpar. PADRAM: Parametric design and rapid meshing system for complex turbomachinery configurations. In *Turbo Expo: Power for Land, Sea, and Air*, volume 8 of *Turbomachinery, Parts A, B, and C*, pages 2135–2148, 06 2012.
- [67] P. Mossier, A. Beck, and D. Munz C. A p-adaptive discontinuous galerkin method with hp-shock capturing. *Journal of Scientific Computing*, 91(1):4, 2022.
- [68] P. Mossier, P. Oestringer, J. Keim, C. Mavriplis, A.D. Beck, and C.D. Munz. Tackling compressible turbulent multi-component flows with dynamic hp-adaptation. *Computers & Fluids*, 306:106928, 2026.

- [69] D. Moxey, R. Amici, and M. Kirby. Efficient matrix-free high-order finite element evaluation for simplicial elements. *SIAM Journal on Scientific Computing*, 42(3):C97–C123, 2020.
- [70] D. Moxey, D. Cantwell C. 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*, 249:107–110, 2020.
- [71] 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, 2017.
- [72] D. Moxey, D. Ekelschot, Ü. Keskin, S.J. Sherwin, and J. Peiró. A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality. *Computer-Aided Design*, 72:130–139, 2016.
- [73] D. Moxey, M. 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.
- [74] 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 International Publishing, Cham, 2015.
- [75] J. Nordström and S. Nikkar. Hyperbolic systems of equations posed on erroneous curved domains. *Journal of Computational Physics*, 308:438–442, 2016.
- [76] G. Ntoukas, G. Rubio, O. Marino, A. Liosi, F. Bottone, J. Hoessler, and E. Ferrer. A comparative study of explicit and implicit large eddy simulations using a high-order discontinuous Galerkin solver: Application to a Formula 1 front wing. *Results in Engineering*, 25:104425, 2025.
- [77] N. Offermans. Aspects of adaptive mesh refinement in the spectral element method, PhD Thesis, KTH. 2019.
- [78] N. Offermans, D. Massaro, A. Peplinski, and P. Schlatter. Error-driven adaptive mesh refinement for unsteady turbulent flows in spectral-element simulations. *Computers and Fluids*, 251:105736, 2023.

- [79] A. S. O’Sullivan, K. Puri, and E. Ferrer. *Best Practices and Mesh Convergence on the Extruded Imperial Front Wing With CharLES Solver*. 2024.
- [80] Khurana P., Liosi A., Sherwin S., Hoessler J., Chatzopoulos A., A. Swift, and F. Bottoni. Industrialization of spectral/hp element method for incompressible, transitional flow around formula 1 geometries. In *AIAA Scitech 2025 Forum*.
- [81] D. Papadogiannis, F. Duchaine, F. Fré, F. Frédé, F. Sicot, L. Gicquel, G. Wang, S. Sté, and S. Moreau. Large eddy simulation of a high pressure turbine stage: Effects of sub-grid scale modeling and mesh resolution. *ASME Turbo Expo*, 2014.
- [82] J.M. Pegrum. *Experimental study of the vortex system generated by a Formula 1 front wing*. PhD thesis, Imperial College London, 2007.
- [83] J Peraire, J Peiró, and K Morgan. Advancing front grid generation. In *Handbook of Grid Generation*, chapter 17. CRC Press, 1999.
- [84] J. Peraire, M. Vahdati, K. Morgan, and O.C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72(2):449–466, oct 1987.
- [85] N. Perrinn. PERRINN F1 Open-Access CAD Model. <https://cad.onshape.com/documents/56d4b1ffe4b0f9d49cca0e99/w/f487040ce072f6f71178e968/e/ebfe374e635ff3310b8d26b9>, 2016. Accessed: 2022-05-03.
- [86] P.O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous galerkin method. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, US, 2006. 44th AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics. AIAA Paper 2006-112.
- [87] P.O. Persson and J. Peraire. Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In *47th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando (FL), USA, 5–9 January 2009. AIAA paper 2009–949.
- [88] T. S. Phillips and C. J. Roy. Residual methods for discretization error estimation. *20th AIAA Computational Fluid Dynamics Conference*, 2011.
- [89] N. Pietroni, M. Campen, A. Sheffer, G. Cherchi, D. Bommers, X. Gao, R. Scateni, F. Ledoux, J. Remacle, and M. Livesu. Hex-mesh generation and processing: A survey. *ACM Trans. Graph.*, 42(2), October 2022.
- [90] Pointwise. <http://www.pointwise.com/>, 2017.
- [91] S.B. Pope. Ten questions concerning the large-eddy simulation of turbulent flows. *New Journal of Physics*, 6:35, 3 2004.

- [92] X. Roca, A. Gargallo-Peiró, and J. Sarrate. Defining quality measures for high-order planar triangles and curved mesh generation. In *20th International Meshing Roundtable*, 2011.
- [93] E. Ruiz-Gironés and X. Roca. Generation of curved meshes for the high-lift common research model. In *AIAA Aviation 2022 Forum*. American Institute of Aeronautics and Astronautics, June 2022.
- [94] E. Ruiz-Gironés, X. Roca, and J. Sarrate. High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation. *Computer Aided Design*, 72:52–64, 2016.
- [95] E. Ruiz-Gironés, J. Sarrate, and X. Roca. Measuring and improving the geometric accuracy of piece-wise polynomial boundary meshes. *Journal of Computational Physics*, 443:110500, 2021.
- [96] L. Karman S. T. Erwin J. S. Glasby R. and D. Stefanski. *High-Order Mesh Curving Using WCN Mesh Optimization*.
- [97] 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.
- [98] J Shewchuk. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. *Applied Computational Geometry: Towards Geometric Engineering*, 1148:203–222, 1996.
- [99] A. Shi and P.-O. Persson. Local element operations for curved simplex meshes. *International Journal for Numerical Methods in Engineering*, 125(2):e7379, 2024.
- [100] H. Si. TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2):11:1–11:36, 2015.
- [101] Siemens Digital Industries Software. Simcenter STAR-CCM+, version 2022.2. <https://www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html>, 2025.
- [102] J. Slaughter, D. Moxey, and S. J. Sherwin. Large eddy simulation of an inverted multi-element wing in ground effect. *Flow Turbulence Combust*, 110:914–944, 2023.
- [103] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts. Robust untangling of curvilinear meshes. *Journal of Computational Physics*, 254:8–26, 2013.

- [104] M. Turner. High-order mesh generation for CFD solvers. *PhD Thesis, Spiral Database*, - Imperial College London, 2018.
- [105] M. Turner, D. Moxey, J. Peiró, M. Gammon, C.R. Pollard, and H. Bucklow. A framework for the generation of high-order curvilinear hybrid meshes for CFD simulations. In *Procedia Engineering*, volume 203, pages 206–218, 2017.
- [106] M. Turner, J. Peiró, and D. Moxey. Curvilinear mesh generation using a variational framework. *Computer-Aided Design*, 103:73–91, 2018. 25th International Meshing Roundtable Special Issue: Advances in Mesh Generation.
- [107] A. Dobrev V. P. Knupp, T. Kolev, K. Mittal, N. Rieben R. M. Stees, and Z. Tomov V. *Asymptotic Analysis of Compound Volume+Shape Metrics for Mesh Optimization*, pages 106–117.
- [108] J.C. Vassberg and A. Jameson. In pursuit of grid convergence for two-dimensional Euler solutions. *Journal of Aircraft*, 47(4):1152–1166, July–August 2010.
- [109] G. Vivarelli, N. Qin, and S. Shahpar. A review of mesh adaptation technology applied to computational fluid dynamics. *Fluids*, 10(5), 2025.
- [110] J. Wallwok. Mesh adaptation and adjoint methods for finite element coastal ocean modelling. *PhD Thesis, Spiral - Imperial College London*, 2021.
- [111] H. Wüstenberg. Implicit time-stepping for scale-resolved industrial flow simulations using spectral/hp elements, PhD Thesis, Spiral - Imperial College London, Feb 2025.
- [112] M. Zhang, C. Wang, S. Kramer, G. Wallwork J. S. Li, J. Liu, X. Chen, and D. Piggott M. Towards universal mesh movement networks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 14934–14961. Curran Associates, Inc., 2024.
- [113] X. Zou, B. Lo S. R. Sevilla, O. Hassan, and K. Morgan. The generation of tetrahedral meshes for nurbs-enhanced fem. *Engineering with Computers*, 40(6):3949–3977, 2024.