University of Southampton

Faculty of Engineering and the Environment
Institute of Sound and Vibration Research

Mesh Curving for Acoustic Simulations with Limited Geometric Knowledge

Verena Stephanie Ziel

Thesis for the degree of Doctor of Philosophy

February 2020

University of Southampton

Abstract

Faculty of Engineering and the Environment
Institute of Sound and Vibration Research

Doctor of Philosophy

Mesh Curving for Acoustic Simulations with Limited Geometric Knowledge

by Verena Stephanie Ziel

High-order simulation techniques are advantageous for acoustic simulations. To effectively apply these methods, the domain geometry has also to be accurately described with high-order elements. In this thesis, mesh curving algorithms are considered under the restriction that only a fine linear target mesh is provided as input geometry. This situation can arise especially in the industrial context, where the original CAD data is not available, e.g. with scanned data or for a subcontracted simulation company.

Here, four mesh curving algorithms are described, one nodal method and three modal methods. Their applicability and curving accuracy is assessed and compared on basic geometries. This leads to a preselection of two modal methods which are then further tested for their influence on the simulation results for Helmholtz scattering problems. A modal curving that is based on the H^1-seminorm optimisation is selected as the more beneficial approach to curve meshes for acoustic simulations. It significantly reduces the geometrically induced field error compared to the other curving approaches. The chosen H^1 modal method is extended to 3D and applied to an academic and a realistic test case.

The second aspect of the thesis is the evaluation of the relation between the geometry discretisation error (GDE) and the field error that is induced by the geometric inaccuracy (GIE). This is first studied for the 2D Helmholtz scattering by a cylinder with nodal meshes obtained with the software *Gmsh*. Different measures are considered for the geometric accuracy and for the field error. The final model is described by an area based GDE and a field error evaluation along a ring in the simulation domain. It shows a linear relation between the GIE and GDE and a super-linear dependency of the frequency ω. Tests with modally curved meshes on the circular geometry and for the scattering by a distorted ellipse show that the considered GDE measure does not fully explain the dependency of the GIE on the geometric accuracy.

Contents

List of Tables

List of Figures

Declaration of Authorship

I, Verena Ziel, declare that this thesis, entitled *Mesh Curving for Acoustic Simulations with Limited Geometric Knowledge*, and the work presented in it are my own and has been generated by me as the result of my own original research. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- Where I have consulted the published work of others, this is always clearly attributed;
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- Parts of this work have been published as:

V.S. Schmid, H. Bériot, O. Atak, G. Gabard, High-order curved mesh generation by using a fine linear target mesh. In *ECCOMAS Congress 2016 Proceedings*, volume 1, pages 493–503. ECCOMAS, 2016.

V. Ziel, H. Bériot, O. Atak, and G. Gabard. Comparison of 2D boundary curving methods with modal shape functions and a piecewise linear target mesh. *Procedia Engineering*, 203, 2017.

V. Ziel, H. Bériot, O. Atak, and G. Gabard. High-order 2D mesh curving methods with a piecewise linear target and application to Helmholtz problems. *Computer-Aided Design*, 105:2641, 2018.

Signed: .....

Date: .....

Acknowledgements

I wish to express my gratitude to my academic and industrial supervisors Dr. Gwénaël Gabard, Dr. Hadrien Bériot, and Dr. Onur Atak who provided outstanding support throughout the PhD period. Thank you very much for your guidance from specifying the topic at the beginning, all the way to your helpful feedback on finalising the written thesis. Furthermore, I wish to show my appreciation to Dr. Michel Tournour for the organisation of the CRANE project and his input on topic ideas, to Prof. Jeremy Astley for helpful impulses especially on acoustic issues, and to Dr. Alan McAlpine, who took over to organise the final steps of my thesis submission and examination. I am indebted to my examiners, Prof. Ruben Sevilla and Prof. Alec Wilson for their constructive criticism and suggestions. The same applies to the anonymous reviewers of the papers whose revision remarks improved not only the papers but also this thesis.

I want to thank my colleagues and friends in Leuven and Southampton for all those little talks and evenings and weekends out that are so important to keep you sound throughout your PhD-time. Special thanks go to Anderson and his family who let me stay with them whenever I came to Southampton after moving back to Germany.

Of course, I want to express my deepest gratitude to my family, especially to my parents who supported me throughout my whole education, to Florian who always believes in me and my dreams and proofread this thesis along the way, and to Nele, who joined us towards the end of my PhD-time. Thank you for bringing so much joy into my life and for showing me how much can be achieved simply by being curious, one step at a time.

Last but not least, I would like to thank the European Union for funding for this thesis as part of the CRANE project (Community and Ramp Aircraft NoisE, www.crane-eid.eu, GA: 606844) under the Framework Programme 7, as well as the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, for access to the computing resources used to complete this work.

Acronyms

APU	Auxiliary power unit.
BC	Boundary conditions.
BEM	Boundary element method.
CAD	Computer aided design.
CFD	Computational fluid dynamics.
CMF	Continuous Moving Frames.
CRANE	Community and ramp aircraft noise.
DGM	Discontinuous Galerkin method.
DoF	Degree of freedom.
FDM	Finite difference method.
FEM	Finite element method.
GDE	Geometry discretisation error.
GIE	Geometry induced error.
IGA	Isogeometric analysis.
NEFEM	NURBS-enhanced FEM.
NURBS	Non-uniform rational B-splines.
PML	Perfectly matched layer.
SPL	Sound pressure level.
WALF	Weighted Averaging of Local Fittings.

Nomenclature

$\boldsymbol{\alpha}_i$	Coefficients of the Lobatto interpolation. (Sec. 3.2)
β_i	Parameter of the model fitted to GDE and GIE, $i = 1, 2, 3$. (Sec. 5.4)
$\boldsymbol{\gamma}$	Multi-index. (Sec. 2.1)
Γ	Boundary of the computational domain Ω. (Sec. 2.2)
$\delta_F(C_1, C_2)$	Fréchet distance of the two curves C_1 and C_2. (Sec. 2.2)
$\delta_H(S_1, S_2)$	(Two-sided) Hausdorff distance of the two sets S_1, S_2. (Sec. 2.2)
$\delta_{i,j}$	Kronecker delta. (Sec. 2.1)
$\boldsymbol{\zeta}$	Mapping $\boldsymbol{\zeta} : \Omega \to \hat{\Omega}$ that maps point $\boldsymbol{x} \in \Omega$ to its position $\hat{\boldsymbol{x}} \in \hat{\Omega}$, approximated by the geometric discretisation. (Sec. 2.2)
Θ	Polar angle. (Sec. 7.2)
$\boldsymbol{\kappa}^{e_i}$	Term in the blending function that corresponds to edge e_i. (Sec. 4.3)
$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$	Triangular barycentric coordinates. (Sec. 2.1)
$\boldsymbol{\xi} = (\xi, \eta) = (\xi_1, \xi_2)$	Point in 2D reference coordinates. (Sec. 2.1)
ρ_0	Ambient mass density field. (Sec. 4.1)
ϕ	Velocity potential. (Sec. 7.1)
ϕ_i	Kernel functions. (Sec. 2.1)
φ	Test function, applied to obtain weak formulation. (Sec. 4.1)
$\varphi_Q^{v_i}$, $\varphi_{Q,q_1}^{e_i}$, $\varphi_{Q,q_1,q_2}^{b_i}$	Vertex, edge, and bubble Lobatto shape functions for quadrangular elements. (Sec. 2.1)
$\varphi_T^{v_i}$, $\varphi_{T,k}^{e_i}$, φ_{T,q_1,q_2}^{b}	Vertex, edge, and bubble Lobatto shape functions for triangle elements. (Sec. 2.1)
$\varphi_{\mathrm{Tet}}^{v_i}$, $\varphi_{\mathrm{Tet},k}^{e_i}$, $\varphi_{\mathrm{Tet},q_1,q_2}^{b}$	Vertex, edge, and bubble Lobatto shape functions for tetrahedral elements. (Sec. 6.3)
Ω	Physical domain, corresponds to computational domain. (Sec. 2.2)
ω	Angular frequency. (Sec. 4.1)
A	Area between $\mathcal{G}_{\text{exact}}$ and $\mathcal{G}_{\text{approx}}$. (Sec. 2.2)
$\mathcal{C}$	Computation mesh, Cmesh. (Sec. 3.2)
c_0	Speed of sound. (Sec. 4.1)
$\boldsymbol{d}^{e_i}$	Difference between the actual, curved edge and the linear vertex interpolation of edge e_i. (Sec. 4.3)
$\boldsymbol{d}_\Omega$	Difference between identity function and the mapping from the exact domain to the actual simulation domain. (Sec. 2.2)
$\frac{D_0}{Dt}$	Material derivative, $\frac{D_0}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{u}_0 \cdot \nabla$. (Sec. 7.1)

$d = d_x$	(x-part of) the deflection of the target curve from the linear interpolation of the vertices. (Sec. 3.2)
$\mathtt{d_e}$	Dimension of the element. (Sec. 2.1)
$\mathtt{d}$	Dimension of the physical space. (Sec. 2.1)
$\mathcal{E}_\mathcal{C}$	Set of $\mathcal{C}$ edges. (Sec. 3.2)
$\mathcal{E}_\mathcal{M}$	Set of $\mathcal{M}$ edges. (Sec. 3.2)
$\tilde{E}_{2,M}$, $\tilde{E}_{2,R}$, $\tilde{E}_{2,P}$	GIE, based on $E_{2,M}$, $E_{2,P}$, and $E_{2,R}$, respectively. (Sec. 5.4)
$E_{2,M}$	Mapping based field error. (Sec. 5.3)
$E_{2,P}$	Projection based field error. (Sec. 5.3)
$E_{2,R}$	Field error evaluated along a ring. (Sec. 4.4)
$\hat{\boldsymbol{f}}(\boldsymbol{\xi})$	Polynomial approximation of curve $\boldsymbol{f}(\xi)$. (Sec. 3.2)
$\boldsymbol{F}$	Mapping $\hat{T} \mapsto T$. (Sec. 2.1)
$\mathcal{G}_{\text{approx}}$	Approximate geometry. (Sec. 2.0)
$\mathcal{G}_{\text{exact}}$	Exact geometry. (Sec. 2.0)
GDE_A	Area based measure of the GDE (geometry discretisation error). (Sec. 3.3)
GDE_H	Hausdorff distance as GDE. (Sec. 5.2)
GDE_M	Domain mapping based GDE. (Sec. 5.2)
GDE_V	Volume based measure of the GDE (geometry discretisation error), generally evaluated towards the target curve $\mathcal{M}$. (Sec. 6.1)
GDE_V^S	Special case of the volume based measure of the GDE (geometry discretisation error), evaluated to the exact Sphere. (Sec. 6.1)
$\text{GDE}_{A_\mathtt{Q}}$	Area based GDE, based on quadranglewise area calculation. (Sec. 3.3)
h	Typical element size in the mesh. (Sec. 1.2)
H^1	Sobolev space, common choice of V. (Sec. 4.1)
$J_{\boldsymbol{F}}$	Jacobian of the mapping $\boldsymbol{F}$. (Sec. 2.4)
k	Acoustic wavenumber, $k = \frac{\omega}{c_0}$. (Sec. 1.2)
kh	Non-dimensional wavenumber. (Sec. 4.1)
$L^2 = L^2(\Omega)$	Space of functions that are quadratically integrable over Ω. (Sec. 4.1)
$l_\mathcal{M}$	Total length of $\mathcal{M}$. (Sec. 3.3)
l_i	i^{th} Lobatto polynomial. (Sec. 2.1)
L_i^n	i^{th} Lagrange basis polynomial of order n, correponds to i^{th} Lagrange line shape functions of order n. (Sec. 2.1)
$L_\mathtt{p}$	Sound pressure level (SPL). (Sec. 7.2)
$\mathcal{M}$	Modelling mesh, Mmesh. (Sec. 3.2)
$\widetilde{\text{MpC}}$	Ratio $\frac{n_\mathcal{M}^e}{n_\mathcal{C}^e}$ (2D) or $\frac{n_\mathcal{M}^f}{n_\mathcal{C}^f}$ (3D). (Sec. 3.3)
MpC	Number of Mmesh edges per Cmesh edge. (Sec. 3.3)
$\boldsymbol{n}$	Outwards facing unit normal of the domain Ω. (Sec. 4.1)
N_i^e	i^{th} local field shape functions in FEM definition. (Sec. 4.1)
N_i	i^{th} global field shape functions in FEM definition. (Sec. 4.1)
$n_\mathcal{C}^e$	Number of $\mathcal{C}$ edges. (Sec. 3.2)
$n_\mathcal{C}^f$	Number of $\mathcal{C}$ faces (in 3D). (Sec. 6.1)
$n_\mathcal{C}^v$	Number of $\mathcal{C}$ vertices. (Sec. 3.2)

n_N	Number of global shape functions N_i, correponds to number of nodes in the mesh. (Sec. 4.1)
$n^e_{\mathcal{M}}$	Number of $\mathcal{M}$ edges. (Sec. 3.2)
$n^f_{\mathcal{M}}$	Number of $\mathcal{M}$ faces (in 3D). (Sec. 6.1)
$n^v_{\mathcal{M}}$	Number of $\mathcal{M}$ vertices. (Sec. 3.2)
n	Order of polynomial basis. (Sec. 2.1)
$N^n_{Q,i}$	i^{th} Lagrange quadrangle shape functions of order n. (Sec. 2.1)
$N^n_{T,i}$	i^{th} Lagrange triangle shape functions of order n. (Sec. 2.1)
$\mathbb{P}^n$	Polynomial space, span by triangle shape functions of order n. (Sec. 2.1)
$\mathtt{p}$	Acoustic pressure field. (Sec. 4.1)
$\mathtt{p}_h$	Piecewise linear approximation of $\mathtt{p}$. (Sec. 4.1)
$\mathtt{p}_{\text{ref}}$	Reference pressure to calculate the sound pressure level, $\mathtt{p}_{\text{ref}} = 2 \cdot 10^{-5}$Pa. (Sec. 7.2)
$\mathtt{p}_{\text{ex}}$	Exact solution of the acoustic pressure field. (Sec. 4.1)
$\tilde{\mathbb{P}}^n$	Polynomial space, span by quadrangle shape functions of order n. (Sec. 2.1)
p	Field interpolation order. (Sec. 1.2)
$P_{Le,i}$	i^{th} Legendre polynomial. (Sec. 2.1)
$\mathtt{q}$	Volume velocity source. (Sec. 4.1)
q	Geometric interpolation order. (Sec. 1.2)
$\mathtt{R}^2$	$\mathtt{R}^2$-measure to evaluate the model fit. (Sec. 5.4)
$\tilde{\mathtt{R}}^2$	Adjusted $\mathtt{R}^2$. (Sec. 5.4)
$\hat{T}$	(General) reference element. (Sec. 2.1)
$\hat{T}^q_{\text{line}}$	Line reference element of order q, $\hat{T}^q_{\text{line}} = [-1, 1]$. (Sec. 2.1)
$\hat{T}^q_{\text{quad}}$	Quadrilateral reference element of order q, $\hat{T}^q_{\text{quad}} = [-1, 1] \times [-1, 1]$. (Sec. 2.1)
$\hat{T}^q_{\text{tria}}$	Triangular reference element of order q, defined by nodes $\boldsymbol{\xi}_1 = (-1, -1)$, $\boldsymbol{\xi}_2 = (1, -1)$,and $\boldsymbol{\xi}_3 = (-1, 1)$. (Sec. 2.1)
T	Physical element. (Sec. 2.1)
t	Time. (Sec. 4.1)
$\boldsymbol{u}$	Acoustic velocity. (Sec. 7.1)
$\mathcal{V}_{\mathcal{C}}$	Set of $\mathcal{C}$ vertices. (Sec. 3.2)
$\mathcal{V}_{\mathcal{M}}$	Set of $\mathcal{M}$ vertices. (Sec. 3.2)
V	Solution space. (Sec. 4.1)
V_h	Finite-dimensional subspace of V, common choice is V^1_h, the space of continuous, piecewise linear functions. (Sec. 4.1)
V^1_h	Space of continuous, piecewise linear functions. (Sec. 4.1)
V^p_h	Space of continuous, piecewise polynomial functions of order p. (Sec. 4.1)
V_{approx}	Volume between the approximation and the target surface. (Sec. 6.1)
$\boldsymbol{x} = (x, y) = (x_1, x_2)$	Point in 2D physical coordinates. (Sec. 2.1)

Chapter 1

Introduction

1.1 General Context

Among the environmental effects of air traffic, the noise from landings and take-offs is the most directly noticeable ([56]). It reduces the quality of life in neighbouring areas, leading to decreases in rents and depreciation of houses. Based on the data from the Amsterdam Airport Schiphol in 1999 as an example, the social cost arising therefrom is estimated between €400 and €900 per landing ([58]). But noise pollution from aircraft is not only triggering annoyance in residents close to airports. It is also seen as a possible risk factor for hypertension ([74]), to have an adverse impact on reading and speech perception of school children ([41]) and is also associated with sleep disturbance and tinnitus ([95]).

Besides this *community noise*, airport staff is also exposed to high noise levels, which is described as *ramp noise*. The main contribution to the ramp noise is emitted by the auxiliary power unit (APU).

The increasing awareness of the negative effects of noise pollution and rise in air traffic led the European Commission to announce an objective of a 65% decrease in aircraft noise from 2000 to 2050 ([25]). This represents a challenge for aircraft and engine manufacturers, which can only be attained by developing breakthrough design innovations and novel acoustic treatments. A design process including prototyping and physical testing is very time consuming and costly. Therefore, advanced virtual prototyping tools, including CAD software to model the geometry and numerical simulations to test the resulting acoustic properties, are required. The complexity of the problem necessitates trade-offs between the accuracy of the solution and the runtime and memory needed to achieve it.

The *CRANE* project, within which this PhD is embedded, aims at reducing this trade-off by developing methods that improve the accuracy that can be reached within reasonable runtime and memory bounds. The main focus thereby lies on the efficient prediction of community and ramp noise.

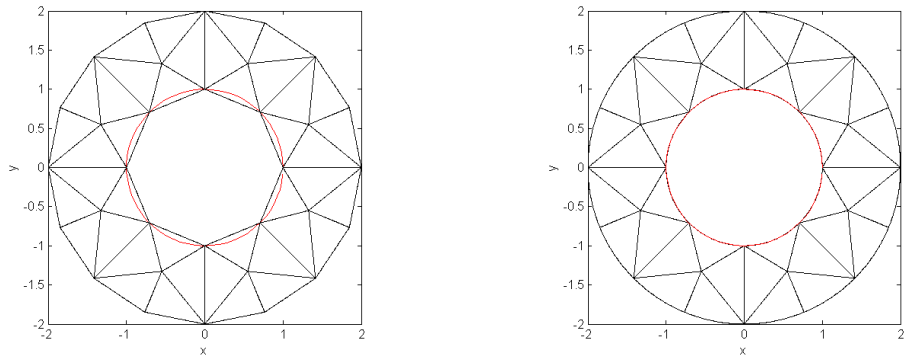
1.2 Motivation for Geometric Enhancement in Acoustical Simulations

Numerical methods are based on the concept of discretisation. In the framework of this PhD, these are spatial discretisation and the discretisation of the field. For computations in the time-domain also time discretisation is needed. In the frequency-domain, the bottleneck for the refinement of the discretisation is given by the available memory. In the time-domain, computations are less memory-consuming and this bottleneck is not present. Therefore, time discretisation is neglected in this thesis and it is focused on simulations in the frequency-domain.
For spatial discretisation, the computational domain can be divided into non-overlapping elements forming a mesh. Some methods do not rely on a mesh structure, but on function evaluations at specified points in the computational domain (the so-called *collocation methods* or *meshless methods*). This PhD will focus on the analysis of mesh-based methods.
In the standard finite element method (FEM), the field is discretised on the vertices of the mesh using elementwise interpolation. To refine the field discretisation, e.g. to avoid large discretisation errors, and thus increase the accuracy of the method, the spatial discretisation has to be refined by introducing more mesh vertices. This amounts to decreasing the typical size h of the elements. This refinement strategy is called h*-refinement*. With h-refinement alone, simulations become prohibitively expensive as the frequency increases (see Equation (4.7)).
To circumvent this, another refinement strategy called p*-refinement* can be applied, with p denoting the interpolation order of the field. When applying p-refinement, the representation of the field is enhanced within individual elements. Methods that implement field representation enhancement are called *high-order methods*. The way the additional information is introduced depends on the method. In most methods, the enhanced field interpolation does not affect the element geometric description, thus the geometric representation remains linear.

In [9], the performance of p-refined FEM with Lobatto polynomial shape functions is tested for fixed field order p. The relation between h, p, and accuracy is evaluated. For this, a three-dimensional simulation of a plane wave with wavenumber k propagating in a cube of size a is considered. A significant reduction of the memory requirements is observed as the simulation order increases. To achieve an error of 1% for simulations with Helmholtz number $ka = 50$, about 30 Gb are needed for a cubic simulation. For $p = 10$, the same accuracy can be achieved with only 2 Gb of memory. The simulation runtimes show a similar improvement.
As the field interpolation order p is increased, the target accuracy can be achieved with coarser meshes. If the mesh is finer, the solution is over resolved. The computation is slowed down and more memory is needed. Therefore, using coarse meshes is essential to fully benefit from the performance improvements of high-order methods.

Geometric enhancement

Along with a coarse linear geometric description comes a new source of error. Computational domains in numerical acoustics involve often curved and complex boundaries. Such structures are only poorly represented by coarse linear interpolation, as could be seen already for a simple



(a) (b)

Figure 1.1: Coarse geometry discretisation of a domain with a circular boundary, discretised by 8 (a) linear and (b) quadratic line elements. The exact boundary is shown in red.

circle in Figure 1.1(a). To alleviate this issue, the geometric interpolation order q could be enhanced as well. Comparing Figures 1.1(a) and 1.1(b), the improvement of the boundary description from linear to quadratic interpolation can be seen. Note that even though the approximation of the circle improves for increasing q, it will not be described exactly by the piecewise polynomial interpolations.

The domain the simulation is typically run on differs from the one that was intended. In other words, the solved problem is different from the problem that was posed. As the accuracy w.r.t. the field representation is increased, the solution converges to the analytical solution of the problem with the approximated domain. The difference between the analytical solutions of the original problem and the actually solved problem is the error induced by the geometric approximation, the *geometry induced error (GIE)*.

1.3 Scope and Objectives

The aim of this PhD project is to decrease the GIE by enhancing the geometric description in acoustic simulations. Therefore, the first part of the thesis covers methods to improve the geometric discretisation. Based on the industrial orientation of the CRANE project, a main focus lies on the applicability of the considered methods to real life problems. Commercial meshing tools typically only provide geometric interpolation up to second order. This is similarly expressed in [45]. The only exception known by the author is the recently developed high-order mesh generation from Pointwise, Inc. [68]. Therefore, enhancement algorithms will be considered that curve a given linear mesh to obtain a better geometric description of the simulation domain.

Most mesh curving algorithms described in the literature rely on the knowledge of the exact geometry, e.g. in form of a parametrisation. However, in many practical situations, the original CAD-geometry is not accessible. In medical applications, only scanned pointwise data is available for patient specific simulations. Similarly, manufactured parts are scanned and the resulting pointwise geometry description is then used for simulations, as the manufacturing process comes along with deviations from the original CAD-geometry. Another example are simulation companies acting as subcontractors, for whom the full CAD data is not provided due

to confidentiality. In this case, the geometric information can be available directly in form of a mesh. For this thesis, it is assumed that only such a fine linear target mesh is available. This could correspond to a mesh used previously for low-order simulations which needs to be coarsened and curved in order to apply it for high-order simulations. Four algorithms are developed that curve a coarse linear mesh towards this refined linear geometry representation. Measures of the *geometry discretisation error (GDE)* will be considered to compare and chose between the methods.
In the literature, several statements are provided about the general link between the geometric accuracy and the accuracy of simulations. Numerical tests have been performed for the tension on a plate with a circular or elliptic hole [54, 55] and for flow around a circular or symmetric obstacle [6], providing mostly qualitative comparisons. The second part of the PhD targets the influence of the geometric accuracy of the domain description onto the accuracy of acoustic simulations. That is, the relation between the GDE and the GIE is analysed for scattering test cases.
Finally, the curving algorithm that proves most advantageous is extended to 3D and the developed method will be applied and tested for a more realistic geometry of a turbofan intake.

The objectives of the PhD are summarised as follows:

- Development of a mesh curving method with only partial information about the original geometry.
- Evaluation of the effect of the GDE on the accuracy of acoustic simulations.
- Application of the method to test cases and a complex geometry.

The outline of this thesis is as follows: In Chapter 2, some background on geometry representation and geometric enhancement methods is given. The developed 2D mesh curving algorithms as well as a standard method from the literature are described in Section 3. An overview is given of GDE measures. As a basis for simulations on the curved meshes, a review of numerical acoustics is given in Chapter 4, as well as how the higher order meshes are embedded into p-FEM, the selected numerical method. In Chapter 5, a literature review on the relation between the GDE and the GIE in general is given. It is followed by a scattering test case on high-order meshes to study the effect of the geometric accuracy in acoustics simulations for both meshes curved with a standard reference approach and meshes that were curved with the developed curving methods. Based on all the tests in the previous chapters, one of the methods is selected and extended to 3D in Chapter 6. As a preparation for realistic geometries, feature line treatment is added. The curving algorithm is then applied for two test cases, including a realistic geometry case of a generic intake in Chapter 7. Chapter 8 summarises the previous chapters and details possible future work.

Chapter 2

Review of Geometric Representation and Enhancement

As mentioned in the Introduction, high-order methods require a coarse mesh definition to operate efficiently (see Section 1.2). At the same time, the geometric interpolation of the mesh remains typically linear. The resulting coarse, straight sided mesh often leads to a poor geometric description of the computational domain.
This section provides a review of the main types of curved geometry representations. An overview of different GDE measures is given, as well as methods to create curved boundaries. Methods to untangle and smooth curved meshes are also reviewed.
To distinguish between the field and geometric approximation, the field interpolation order is denoted by p, the geometric interpolation order by q. The exact geometry is called $\mathcal{G}_{\text{exact}}$, whereas the approximate geometry is called $\mathcal{G}_{\text{approx}}$.

2.1 Geometry Representation

2.1.1 Polynomial Interpolation

The standard way of discretising the computational domain and representing its geometry is by using a mesh composed of linear, straight-sided elements. Linear mesh elements are described by their vertices and by the linear interpolation between those vertex nodes. In this way, boundary curves are represented by a polygon, boundary surfaces for 3D domains by a polyhedral surface.
Alternatively, e.g. high-order polynomial interpolation can be applied. Different polynomial bases come into consideration. In the following, nodal Lagrange bases and modal Lobatto bases will be considered. Independent of the choice of the polynomial basis, interpolations are typically based on parametric reference mappings.

2.1.1.1 Reference Mapping

An important concept in interpolation based mesh description is the parametric mapping from a reference element to the physical element. Thereby, the geometry and location of a single element T of the mesh is described by a mapping $\boldsymbol{F} : \mathbb{R}^{\mathtt{d_e}} \to \mathbb{R}^{\mathtt{d}}$, $\hat{T} \mapsto T$ with $\hat{T}$ a *reference*

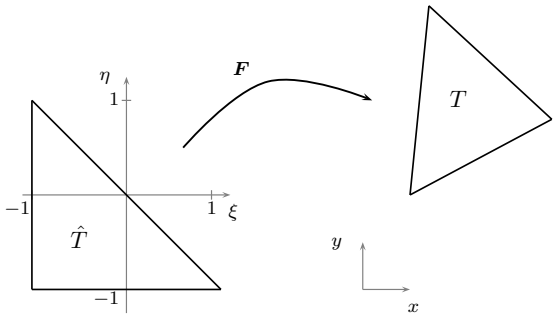
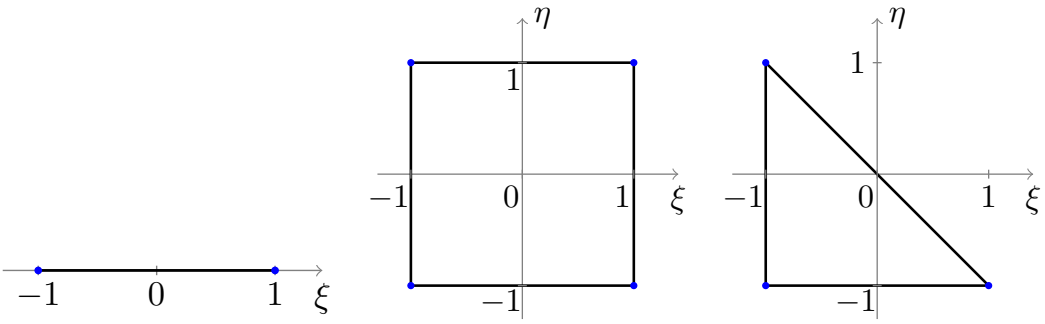


Figure 2.1: Schematic of the mapping $\boldsymbol{F}$ from the reference element $\hat{T}$ to the physical element T.



(a) $\hat{T}^1_{\text{line}}$ (b) $\hat{T}^1_{\text{quad}}$ (c) $\hat{T}^1_{\text{tria}}$

Figure 2.2: Linear Lagrange reference elements.

element, $\mathtt{d_e}$ the dimensions of the element (e.g. $\mathtt{d_e} = 1$ for line elements) and $\mathtt{d}$ the dimensions of the physical space. T is called the *physical element*. The principle is shown in Figure 2.1.

Standard reference elements are the interval $\hat{T}^q_{\text{line}} := [-1, 1]$ for line elements, its Cartesian product $\hat{T}^q_{\text{quad}} := [-1, 1] \times [-1, 1]$ for quadrilateral elements, and the triangle $\hat{T}^q_{\text{tria}}$ defined by the nodes $\boldsymbol{\xi}_1 = (-1, -1)$, $\boldsymbol{\xi}_2 = (1, -1)$, and $\boldsymbol{\xi}_3 = (-1, 1)$ for triangular elements (see Figure 2.2). Hexahedral and tetrahedral 3D reference elements are defined analogously.

2.1.1.2 Lagrange Basis Polynomials

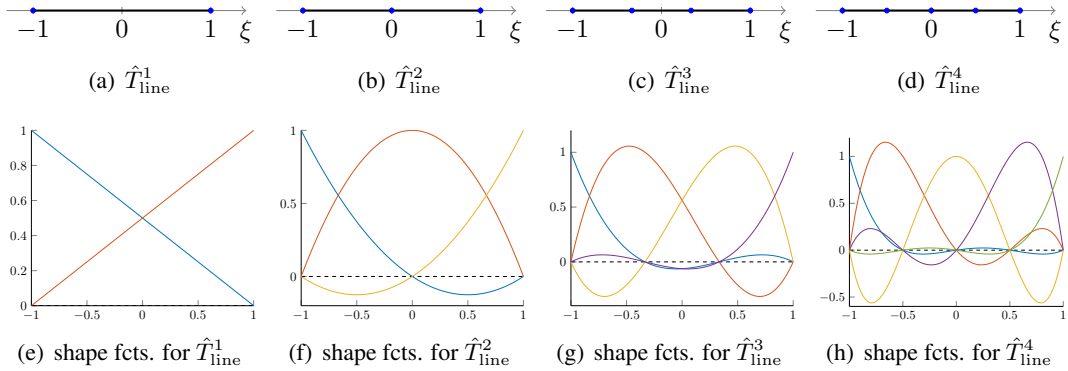
The most common high-order interpolation technique is using a nodal interpolation with Lagrange basis polynomials.

For a 1D interpolation

$$p_n(x) = \sum_{i=0}^{n} y_i L_i^n(x)$$

of order n through nodes $\boldsymbol{x}_i = (x_i, y_i)$, the $n + 1$ basis functions are defined as

$$L_i^n(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j} \right), \quad i = 0, \dots n,$$



(a) $\hat{T}^1_{\text{line}}$ (b) $\hat{T}^2_{\text{line}}$ (c) $\hat{T}^3_{\text{line}}$ (d) $\hat{T}^4_{\text{line}}$

(e) shape fcts. for $\hat{T}^1_{\text{line}}$ (f) shape fcts. for $\hat{T}^2_{\text{line}}$ (g) shape fcts. for $\hat{T}^3_{\text{line}}$ (h) shape fcts. for $\hat{T}^4_{\text{line}}$

Figure 2.3: Line Lagrange reference elements for orders $q = 1, \dots, 4$ and the corresponding shape functions.

and fulfil the property [66]

$$L_i^n(x_j) = \delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} . \tag{2.1}$$

An example of a piecewise quadratic Lagrange interpolation is given in Figure 2.7(b).
In addition to the two vertices, $q-1$ additional nodes $\boldsymbol{x}_i$ are inserted along both the reference and the physical element. *Lagrange line elements* are obtained by placing the nodes in the reference space equidistantly and using Lagrange basis polynomials for these equidistant nodes to define the reference mapping. Examples of high-order line reference elements and the corresponding Lagrange basis polynomials are shown in Figure 2.3. The mapping $\boldsymbol{F}$ is defined as a 1D polynomial interpolation in each dimension:

$$\boldsymbol{F}(\xi) = \sum_{i=1}^{n} \boldsymbol{x}_i L_i^n(\xi).$$

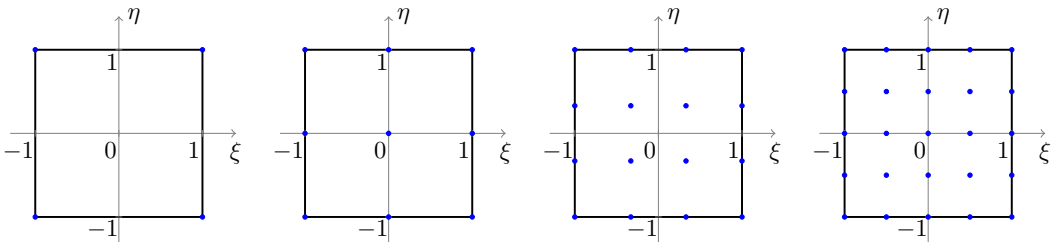
In the context of the element mapping, the polynomials L_i^n are called *shape functions*.

2.1.1.3 Higher-dimensional Lagrange Shape Functions

Analogously to the definition of the Lagrange basis polynomials, shape functions are defined for the 2D and 3D reference elements.
High-order *Lagrange quadrangular* or *hexahedral elements* are defined by tensor products of Lagrange line elements. Accordingly, the interpolation nodes form an equidistant 2- or 3-dimensional lattice on the reference element (see Figure 2.4). Shape functions $N^n_{Q,i}$ for these elements are obtained by multiplying the corresponding 1D shape function for each dimension:

$$N^n_{Q,i}(\boldsymbol{\xi}) = \prod_{k=1}^{\mathtt{d}} L^n_{\beta(i,k)}\left(\xi_k\right),$$

with $d \in \{2, 3\}$, ξ_k the k^{th} coordinate of $\boldsymbol{\xi}$, i the node number in the reference element, and $\beta(i,k)$ the 1D node number of node number i in dimension k. An example of the mapping β is provided in Table 2.1 for the reference element shown in Figure 2.5.



(a) $\hat{T}^1_{\text{quad}}$ (b) $\hat{T}^2_{\text{quad}}$ (c) $\hat{T}^3_{\text{quad}}$ (d) $\hat{T}^4_{\text{quad}}$

Figure 2.4: Quadrangle Lagrange reference elements for order $q = 1, \ldots, 4$.

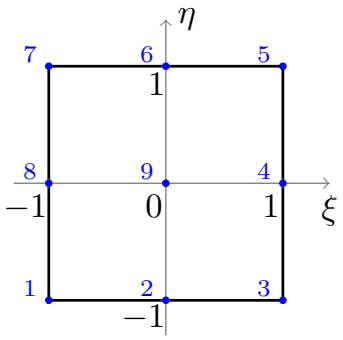


Figure 2.5: Lagrange reference element $\hat{T}^2_{\text{quad}}$ with node numbers in blue.

2D node index i	1	2	3	4	5	6	7	8	9
$\beta(i,1)$	1	2	3	3	3	2	1	1	2
$\beta(i,2)$	1	1	1	2	3	3	3	2	2

Table 2.1: Example of index mapping β, corresponding to the element shown in Figure 2.5.

Based on the construction of the shape functions, they inherit the property (2.1) for interpolation nodes $\boldsymbol{\xi}_j$

$$N^n_{Q,i}(\boldsymbol{\xi}_j) = \delta_{i,j} \tag{2.2}$$

and span the space of polynomials

$$\tilde{\mathbb{P}}^n = \left\{ \sum_j c_j \boldsymbol{\xi}^{\boldsymbol{\gamma}_j} \;\middle|\; \max_i |\gamma_{j,i}| \leq q,\ \gamma_{j,i} \in \mathbb{N}_0 \right\}$$

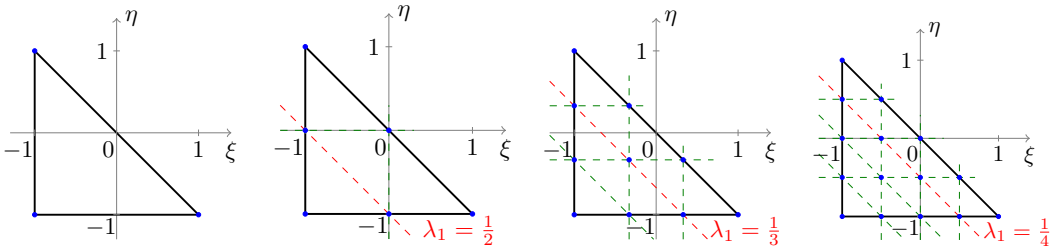
with multi-index $\boldsymbol{\gamma_j} = (\gamma_{j,1}, \ldots, \gamma_{j,\mathtt{d_e}})$ [100].

For triangular or tetrahedral elements, a simple tensor product approach is not suitable. Instead, a barycentric coordinate system is applied to find shape functions $N^n_{T,i}$ that satisfy the property

$$N^n_{T,i}(\boldsymbol{\xi}_j) = \delta_{i,j},$$

analogously to (2.1) and (2.2) for a set of interpolation nodes $\boldsymbol{\xi}_i$ on the triangular or tetrahedral element.

The *barycentric coordinates* λ_i for a point $\boldsymbol{P}$ with Cartesian coordinates (ξ_p, η_p) in a triangle



(a) $\hat{T}^1_{\text{tria}}$ (b) $\hat{T}^2_{\text{tria}}$ (c) $\hat{T}^3_{\text{tria}}$ (d) $\hat{T}^4_{\text{tria}}$

Figure 2.6: Triangle Lagrange reference elements for orders $n = 1, \ldots, 4$.

with vertices (ξ_i, η_i), $i \in \{1, 2, 3\}$ are defined by the equations [100]

$$\xi_p = \sum_{i=1}^{3} \lambda_i(\boldsymbol{P})\xi_i, \quad \eta_p = \sum_{i=1}^{3} \lambda_i(\boldsymbol{P})\eta_i, \quad 1 = \sum_{i=1}^{3} \lambda_i(\boldsymbol{P}).$$

For the reference triangle $\hat{T}_{\text{tria}}$ they can be determined as

$$\lambda_1(\xi, \eta) = -\frac{1}{2}(\xi + \eta), \quad \lambda_2(\xi, \eta) = \frac{1}{2}(\xi + 1), \quad \lambda_3(\xi, \eta) = \frac{1}{2}(\eta + 1). \tag{2.3}$$

Analogously, barycentric coordinates for tetrahedrals can be defined in 3D.

For triangular or tetrahedral Lagrange elements of order n, nodes are placed on the reference element at crossings of the lattice in barycentric coordinates with width $\frac{1}{n}$. Example elements are shown in Figure 2.6. The corresponding shape functions of order n for an interpolation node $\boldsymbol{\xi}_j = \boldsymbol{\lambda_j}$ can be constructed as a product of a scaling coefficient c_i and n terms $z_k(\boldsymbol{\lambda})$

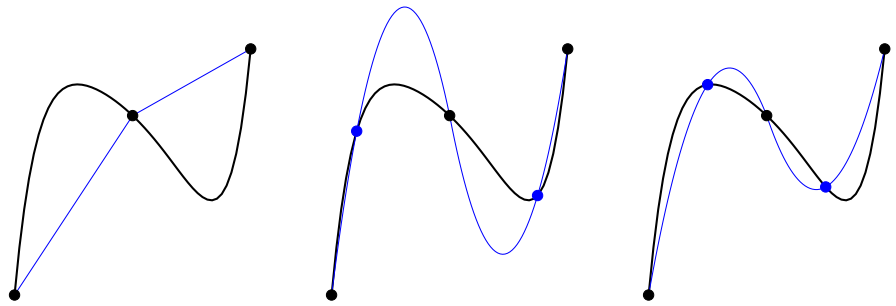
$$N^n_{T,i}(\boldsymbol{\lambda}) = c_i \prod_{k=1}^{n} z_k(\boldsymbol{\lambda}),$$

where each $z_k(\boldsymbol{\lambda})$ vanishes along at least one line of the $\boldsymbol{\lambda}$-lattice that does not cross in the node $\boldsymbol{\xi_j}$. These shape functions form a basis for polynomials of order n:

$$\mathbb{P}^n = \left\{ \sum_j c_j \boldsymbol{\xi}^{\boldsymbol{\gamma_j}} \;\middle|\; \sum_{i=1}^{\mathsf{d_e}} \gamma_{j,i} \leq q,\ \gamma_{j,i} \in \mathbb{N}_0 \right\},$$

with multi-index $\boldsymbol{\gamma_j} = (\gamma_{j,1}, \ldots, \gamma_{j,\mathsf{d_e}})$ [100].

A key parameter is the position of the interior nodes added along the edges (see Figure 2.7(b) and Figure 2.7(c)). Interpolation based on equidistant node spacing as for the Lagrange elements defined above is known to suffer from spurious oscillation for higher interpolation orders [52]. To alleviate this issue, other node placings have been proposed ([85, 96, 40]). Note further that not only the geometric accuracy should be evaluated when considering node positions. The simulation accuracy is not only affected by the error in the geometric discretisation, but also by highly-varying, non-smooth Jacobians of the parametric mapping. Therefore, the Jacobian should be monitored as well, targeting for nodes that are spaced similarly as those in the reference space.



(a) (b) (c)

Figure 2.7: Examples of piecewise (a) linear and (b)-(c) quadratic interpolation (blue line) of a given curve (black line). Black dots indicate the element vertices, blue dots the additional interpolation nodes. Figure (b) and (c) also show the effect of different node locations.

2.1.1.4 Lobatto Shape Functions

Modal polynomial bases provide node-free interpolations. Integrated Legendre polynomials, also called *Lobatto polynomials* form such a modal basis [82]. They are defined by

$$l_k(\xi) := \frac{1}{\|P_{Le,k-1}\|_2} \int_{-1}^{\xi} P_{Le,k-1}(x)\,\mathrm{d}x \quad \text{for } 2 \leq k,\ \xi \in [-1,1]\,,$$

with initial polynomials

$$l_0(\xi) := \frac{1-\xi}{2}, \quad l_1(\xi) := \frac{\xi+1}{2}, \tag{2.4}$$

and $P_{Le,i}(\xi)$ as the i^{th} Legendre polynomial. The Legendre polynomials are recursively defined by

$$P_{Le,k}(\xi) := \frac{2k-1}{k}\xi P_{Le,k-1}(\xi) - \frac{k-1}{k} P_{Le,k-2}(\xi) \quad \text{for } 2 \leq k,$$

with initial polynomials

$$P_{Le,0}(\xi) := 1, \quad P_{Le,1}(\xi) := \xi.$$

Figure 2.8 illustrates the Lobatto polynomial basis functions up to order $q = 5$. Note that the bases are hierarchic. For the Lobatto basis $\mathcal{B}^n$ of order n it holds $\mathcal{B}^{n-1} \subset \mathcal{B}^n$. Therefore, only the additional higher-order shape function and the corresponding coefficients need to be calculated when the order q is changed, in contrast to e.g. the Lagrange polynomial bases, where a change of order requires a recalculation of the full basis. Further it can be noticed in Figure 2.8 that the basis polynomials can be grouped in *vertex functions* l_0 and l_1 that have non-zero contribution at one of the element vertices and correspond to the linear Lagrange polynomial basis, and *bubble functions* l_i for $i \geq 2$ that vanish at the vertices.

The Legendre polynomials are orthogonal in L^2:

$$\int_{-1}^{1} P_{Le,i}(\xi) P_{Le,j}(\xi)\,\mathrm{d}\xi = \begin{cases} \frac{2}{2i+1} & \text{if } i = j \\ 0 & \text{else} \end{cases}.$$

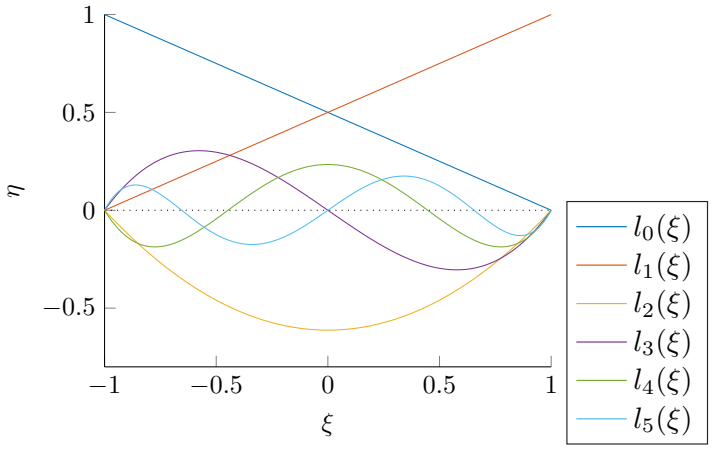
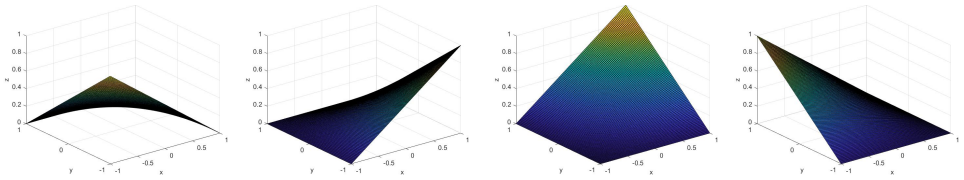


Figure 2.8: Lobatto polynomial basis functions up to order $q = 5$.



(a) $\varphi_Q^{v_1}$ (b) $\varphi_Q^{v_2}$ (c) $\varphi_Q^{v_3}$ (d) $\varphi_Q^{v_4}$

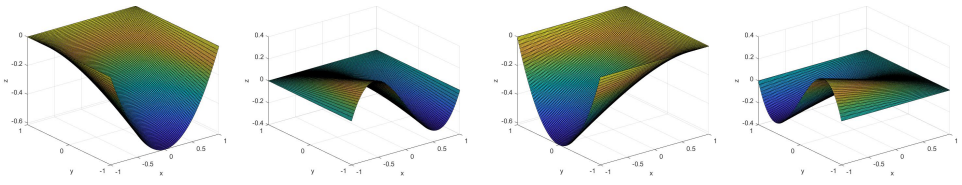
Figure 2.9: Lobatto vertex shape functions for quadrangular elements.

Thus, the Lobatto shape functions fulfil the property

$$\int_{-1}^{1} l_i'(\xi) l_j'(\xi)\,\mathrm{d}\xi = \delta_{ij}, \text{ with } \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}, \text{ for } i, j \geq 2. \tag{2.5}$$

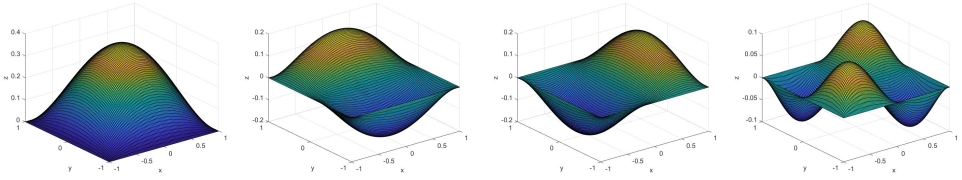
As in the Lagrange case, the definition of modal shape functions for quadrilateral elements is just the product of basis polynomials for the two coordinates.
The product of two 1D vertex functions results in *vertex shape functions* $\varphi_Q^{v_i}$ with non-zero contribution at one of the vertices (see Fig. 2.9). *Edge shape functions* $\varphi_{Q,q_1}^{e_i}$ are obtained by multiplying a 1D vertex function in one dimension with an 1D edge function in the other dimension. The order of the 2D edge shape function corresponds to the order q_1 of the 1D edge function. Its trace along the edge of the quadrilateral element coincides with the 1D edge function. As for the 1D basis functions, 2D edge shape functions vanish at the vertices of the element. Further, each one of them has only non-zero contribution at one of the element edges and vanishes on all three other edges. The edge on which the 2D edge shape function is non-zero is determined by the 1D vertex function (see Fig. 2.10).
The product of two 1D edge functions finally leads to *bubble shape functions* φ_{Q,q_1,q_2}^{b}. These vanish all along the boundary of the element and are non-zero only in the interior of the element (see Fig. 2.11). The order of the bubble shape functions is determined as the maximum of the 1D edge function orders q_1 and q_2.

Shape functions for triangular elements are also grouped in vertex, edge, and bubble shape functions. The linear vertex shape functions are simply defined as the corresponding barycentric



(a) $\varphi^{e_1}_{Q,2}$ (b) $\varphi^{e_1}_{Q,3}$ (c) $\varphi^{e_4}_{Q,2}$ (d) $\varphi^{e_4}_{Q,3}$

Figure 2.10: Lobatto edge shape functions for quadrangular elements of orders $q = 2, 3$ and edges e_1 and e_4.



(a) $\varphi^{b}_{Q,2,2}$ (b) $\varphi^{b}_{Q,3,2}$ (c) $\varphi^{b}_{Q,2,3}$ (d) $\varphi^{b}_{Q,3,3}$

Figure 2.11: Lobatto bubble shape functions for quadrangular elements of orders $q = 2, 3$.

coordinate

$$\varphi^{v_i}_T(\boldsymbol{\xi}) := \lambda_i(\boldsymbol{\xi}).$$

As in the quadrilateral case, they correspond to the linear Lagrange shape functions (see Fig. 2.12).

For the definition of edge shape functions, *Kernel functions* ϕ_i are introduced. They are defined by the decomposition

$$l_k(\xi) = l_0(\xi) l_1(\xi) \phi_{k-2}(\xi) \quad \text{for } k \geq 2,$$

with l_0 and l_1 as defined in (2.4). Triangle edge shape functions of order k are then obtained by

$$\varphi^{e_i}_{T,k}(\boldsymbol{\xi}) := \lambda_i(\boldsymbol{\xi})\lambda_{i+1}(\boldsymbol{\xi})\phi_{k-2}\left(\lambda_{i+1}(\boldsymbol{\xi}) - \lambda_i(\boldsymbol{\xi})\right), \quad i = 1, \ldots, 3, \tag{2.6}$$

with $\lambda_4 := \lambda_1$. Analogously to quadrilateral edge shape functions, they correspond to the trace of l_k on the edge e_i and have zero contribution on the remaining two edges (see Fig. 2.13).

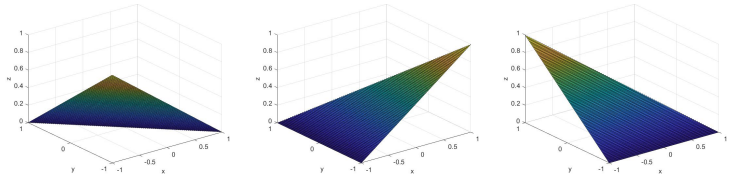
Bubble shape functions that vanish on the element boundary are defined as

$$\varphi^{b}_{T,q_1,q_2}(\boldsymbol{\xi}) := \lambda_1(\boldsymbol{\xi})\lambda_2(\boldsymbol{\xi})\lambda_3(\boldsymbol{\xi})\phi_{n_1-1}\left(\lambda_2(\boldsymbol{\xi}) - \lambda_1(\boldsymbol{\xi})\right)\phi_{n_2-1}\left(\lambda_1(\boldsymbol{\xi}) - \lambda_3(\boldsymbol{\xi})\right),$$

with $n_1, n_2 \geq 1$ (see Fig. 2.14). The order of the bubble shape function is determined as $q = q_1 + q_2 + 1$.

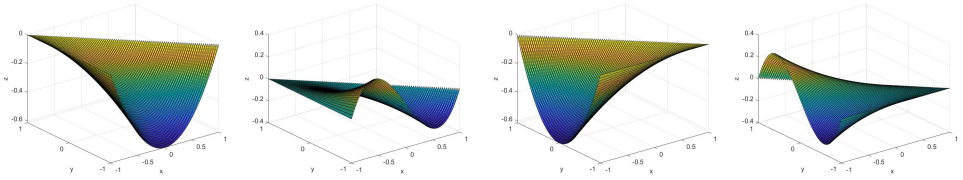
There exists a more intuitive formula, but the one introduced here was found by Solin et al. to have better conditioning properties [82]. By renumbering the barycentric coordinates, an orientation can be added to the element. This is needed for the definition of 3D modal shape functions.

For more information on Lobatto polynomials and Lobatto shape functions see [82].



(a) $\varphi_T^{v_1}$ (b) $\varphi_T^{v_2}$ (c) $\varphi_T^{v_3}$

Figure 2.12: Lobatto vertex shape functions for triangular elements.



(a) $\varphi_{T,2}^{e_1}$ (b) $\varphi_{T,3}^{e_1}$ (c) $\varphi_{T,2}^{e_3}$ (d) $\varphi_{T,3}^{e_3}$

Figure 2.13: Lobatto edge shape functions for triangular elements of orders $q = 2, 3$ and edges e_1 and e_3.

An alternative way to derive a hierarchic modal basis is described in [50].

2.1.1.5 Smooth Representations

The smoothness of the curve can be an important property of the approximation. Even though polynomials are infinitely smooth, the elementwise geometric interpolation leads to a loss of continuity at the element boundaries. With Lagrange or Lobatto polynomials, only a C^0 continuous geometric approximation is obtained. For smoother curves and surfaces, Hermite polynomials can be used.

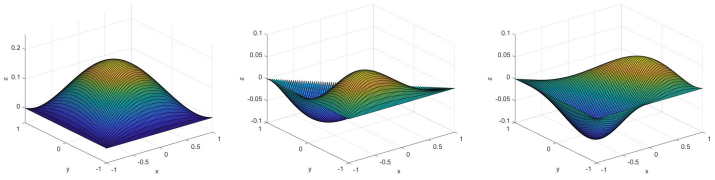
In [18], elements are constructed that allow a G^1 continuous surface. G^1 continuity is a weakened form of C^1 continuity. While the latter depends on the parametrisation of the surface, G^1 continuity only considers the smoothness of the manifold the surface corresponds to. In 2D, a 1D curve is C^1 continuous at a point if the tangents from both sides of the point are equal. For G^1 continuity, the two tangents are still parallel, but of different lengths. Demkowicz et al. illustrate the difference with an example of two parametrisations of the x_3 plane: The normal direction is constant along the plane and therefore the G^1 continuity is obtained by default. With

$$x_1 = \eta_1, \quad x_2 = \eta_1 + \eta_2, \quad x_3 = 0,$$

a C^1-continuous parametrisation is provided. But the parametrisation

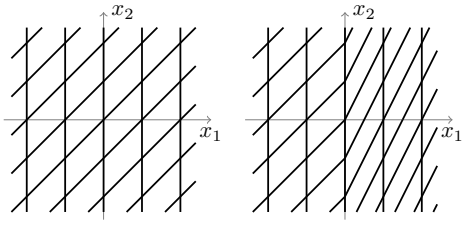
$$x_1 = \eta_1, \quad x_2 = \begin{cases} \eta_1 + \eta_2 & \text{for } \eta_1 \leq 0 \\ 2\eta_1 + \eta_2 & \text{for } \eta_1 > 0 \end{cases}, \quad x_3 = 0$$

is not differentiable at $\eta_1 = 0$ and therefore not C^1-continuous. Note that the equations presented here are corrected versions of those in [18]. To visualise this example, a plot of the coordinate lines is provided in [18] and reproduced in Figure 2.15.



(a) $\varphi^b_{T,1,1}$ (b) $\varphi^b_{T,2,1}$ (c) $\varphi^b_{T,1,2}$

Figure 2.14: Lobatto bubble shape functions for triangular elements of orders $q = 3, 4$.



(a) C^1 continuity (b) G^1 continuity

Figure 2.15: Plot of the coordinate lines of the example parametrisations to visualise the difference between C^1 and G^1 continuity. This figure is based on a figure in [18].

2.1.2 NURBS-based Interpolation

Besides polynomials, another class of basis functions, namely the *Non-uniform rational B-splines (NURBS)*, can be used to represent curved edges and surfaces. To represent a given geometry, several NURBS curves (2D) or patches that are quadrangular in the parametric space (3D) are typically required. Within one curve, the parametrisation changes at *breakpoints*, where the continuity order can be decreased down to C^0 to allow for sharp corners within the curve. NURBS are able to represent many geometries exactly and are a standard way to describe geometries in CAD applications ([79, 80]).

2.2 Quantification of the Geometric Accuracy

In the literature, several measures have been proposed to quantify the geometric error, characterised by the difference between $\mathcal{G}_{\text{approx}}$ and $\mathcal{G}_{\text{exact}}$.

2.2.1 Distance-based Measures

In [2] the maximum of the radial distance between a sphere and the approximating surface is evaluated. This pointwise evaluation provides a simple measure for the GDE, but is very specific for the sphere case. For other geometries, the choice of corresponding points on the exact and the approximating curve or surface becomes arbitrary, but has an effect on the resulting measure.
A more general measure of the difference between two curves is the Hausdorff distance

$$\delta_H(S_1, S_2) = \max\left(\tilde{\delta}_H(S_1, S_2), \tilde{\delta}_H(S_2, S_1)\right), \text{ with } \tilde{\delta}_H(X, Y) = \sup_{x \in X} \inf_{y \in Y} d(x, y), \tag{2.7}$$

for bounded $S_1, S_2 \subset \mathbb{R}$ and a metric $d(\cdot, \cdot)$ (see [3]). It is used as the minimised distance in polygon matching [3] and as a deviation measure in simplification of meshes [37] or

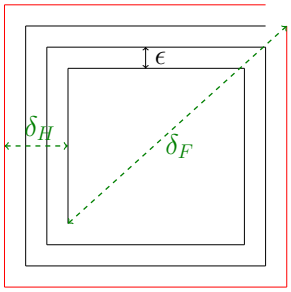


Figure 2.16: Example curves to show the difference between the Hausdorff and the Fréchet distance. This figure is based on the figure in [72].

point-sampled surfaces [65]. It is a common measure for the distance between two sets. For example, S_1 could be the set of blue houses in a city and S_2 the set of green houses respectively. All people living in a blue house are assigned the task to visit one arbitrary person living in a green house and vice versa. Assume that all persons go to the person from the other group that lives the closest to their own place. In the end, the Hausdorff distance between the blue and green houses is given by the longest (one way) distance that was walked by any of the persons to fulfil their task.

Another popular distance-based measure to compare two curves is the Fréchet distance. It is often explained by the analogy of a man walking with his dog ([3, 94]). While the man walks along curve C_1, the dog's path is described by curve C_2, where both never walk backwards. The dog's leash connecting them always follows the shortest connecting line between them and in the end has to have at least the length of the maximal distance which occurred during the walk. The outcome depends on the relative speed and speed variation of the man and the dog w.r.t. each other. Considering all possible walking behaviours, the Fréchet distance between the curves corresponds to the minimal needed length of the dog's leash. Mathematically, it is given as

$$\delta_F(C_1, C_2) = \inf_{\alpha} \left(\max \left(d\left(\beta_{C_1}(s), \alpha(s)\right) | s \in [0, l_{C_1}] \right) \right), \text{ with } \beta : [0, l_{C_1}] \to \mathbb{R}^2,$$

where β_{C_1} is the parametrisation of C_1 by arc length and all possible injective parametrisations of C_2 w.r.t. the length l_{C_1} of C_1 are covered by $\beta : [0, l_{C_1}] \to \mathbb{R}^2$ [3]. There is no need to compute the infimum over possible parametrisations of C_1. Recall that the needed leash length only depends on the relative speed. If both, man and dog just walked twice as fast, the needed length would be the same.
In [3] it is shown that for two closed, convex curves, the Hausdorff distance and the Fréchet distance are the same. In other cases this is not necessarily true. An example is given in Figure 2.16. The Hausdorff distance between the two curves is 3ϵ and thus vanishes as $\epsilon \to 0$, where as the Fréchet distance converges to the length of the diagonal of the square indicated by the red curve [72]. This example also shows that the Fréchet distance is consistent with the intuitive understanding of the difference between two curves.
In [59], an algorithm for a discrete Fréchet distance between two polygonal curves is introduced. The bound on the error between the analytical and the discrete Fréchet distance provided in Theorem 1 of the same paper shows a linear dependency on the distance between the curve sampling points. Looking at applications where a high accuracy is needed and the given runtime

of $\mathcal{O}(m_1 m_2)$ for m_1 and m_2 sampling points on the two curves, the Fréchet distance remains difficult to obtain, as also discussed in [72].
Note that both, Hausdorff and Fréchet distance generally depend on the metric $d(\cdot, \cdot)$. In all cases considered here, $d(\cdot, \cdot)$ is the Euclidean metric.

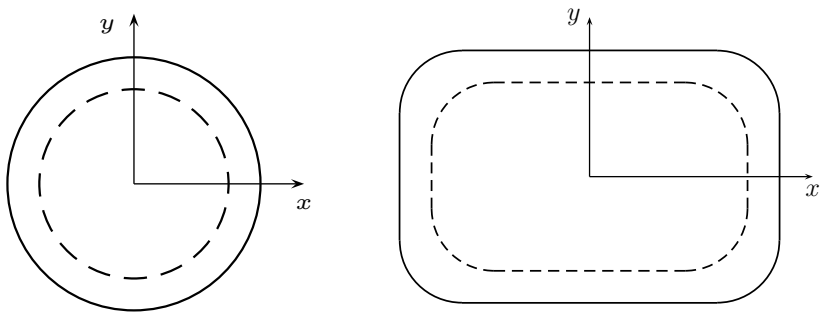
A variation of the Fréchet distance was introduced in [75]. Instead of taking infimum of the maximum or supremum over the pointwise distances between the two curves, the infimum of the L^2 norm of the pointwise distance is considered. The resulting measure is continuously differentiable which is applicable in optimisation based mesh curving (see Section 2.3).

2.2.2 Area-based Measure

As an alternative to distance-based measures of the GDE, [72] considers a measure based on the area A between $\mathcal{G}_{\text{exact}}$ and the $\mathcal{G}_{\text{approx}}$. The area is calculated segmentwise, where the segments are bounded by the intersection points of the two curves. In each segment, the curves are approximated by polygons, using a de Casteljau subdivision scheme. The Bézier curve representation of $\mathcal{G}_{\text{approx}}$ is considered, because the convex hull property provides an upper limit estimate of the accuracy of the polygonal approximation. For most CAD-based $\mathcal{G}_{\text{exact}}$, the same principle can be applied. Where no Bézier curve representation is available, the polygon nodes are sampled recursively to ensure a predefined accuracy. The area between the polygonal approximations is triangulated to evaluate as an approximation of the segment area. The sum over all segment areas is then divided by the arithmetic mean of the length of the two bounding polygons. Numerical tests show a relation between the two types of measures: Optimisation w.r.t. the area based measure also leads to a decrease in the Hausdorff distance between the two curves [72].

2.2.3 Mapping-based Measure

In [97], a measure of the GDE is constructed that takes into account not only the curve describing the boundary Γ, but the whole domain Ω. The paper stresses the difference between the real physical domain Ω, which is represented by $\mathcal{G}_{\text{exact}}$, and the approximated physical domain $\hat{\Omega}$ used for the simulation. Motivated by this discrepancy, a mapping $\boldsymbol{\zeta} : \Omega \to \hat{\Omega}$ is constructed which maps a point $\boldsymbol{x} \in \Omega$ to its position $\hat{\boldsymbol{x}} \in \hat{\Omega}$, approximated by the geometric discretisation. For the GDE measure, the $H^1(\Omega)$ seminorm of the function $\boldsymbol{d}_\Omega(\boldsymbol{x}) := \mathrm{id}(\boldsymbol{x}) - \boldsymbol{\zeta}(\boldsymbol{x})$ is divided by the $H^1(\Omega)$ seminorm of the identity function $\mathrm{id}(\boldsymbol{x}) = \boldsymbol{x}$.
Note that for this definition of the GDE, the shape of interior elements is also considered. In the case of a sphere as in [97], a linear mapping in polar coordinates can be used to describe $\mathcal{G}_{\text{exact}}$. To be without contribution to the GDE, elements in the domain interior need to be described by a linear mapping in polar coordinates. In [97], the mesh seems to be adapted to follow this requirement. For real-life geometries, there is generally no such simple mapping which could be used to adapt interior elements accordingly and, in standard curved meshes, interior elements are mostly straight sided with respect to Cartesian coordinates, independently of the accuracy of the boundary representation.



(a) (b)

Figure 2.17: Schematics of geometries where the differences of (a) the (normalised) normals or (b) the curvatures are zero for carefully chosen parametrisations.

2.2.4 Differential Geometry-based Measures

Differential geometric properties of $\mathcal{G}_{\text{exact}}$ and $\mathcal{G}_{\text{approx}}$ could also be used to define measures of the GDE. The maximal angle between the normals of the exact and the approximated surface are considered in [2] to evaluate the accuracy of a polynomial fitting with different quadrangle element types to the unit sphere.

In computer vision, curvatures such as the Gaussian and the mean curvature of a point on a the surface are of special interest. They are the product or mean of the principal curvatures respectively, which are the minimal and maximal directional curvatures of the surface at this point [57]. In [57], the mean and standard deviation of the Gaussian curvature and the mean curvature, as well as those of the angle between the exact and the approximated normal were measured to compare different local geometry-fitting methods.

The normal, Gaussian and mean curvature are pointwise measures. Therefore, the described measures are dependent on the parametrisations of the curves. A further disadvantage is that these measures are not strictly proper, meaning that they can be zero for non-perfect fits. For the normal, an example are two circles with different radii and parametrisations by the rotation angle (see Figure 2.17(a)). For the curvature in 2D, two quadrangles with rounded corners can be considered (see Figure 2.17(b)). Along the straight sided edges, the curvature is zero. The curved corners are described by a quarter of a unit circle and therefore have a curvature of 1. The parametrisations have to be chosen in such a way that the beginnings and ends of the rounded corners are reached for the same parameters.

2.3 Creation of Curved Boundary Elements

Commonly, high-order curvilinear meshes are generated by applying a-posteriori curving methods. The process begins with a straight sided linear mesh, which is then curved to match the initial geometrical boundary $\mathcal{G}_{\text{exact}}$. An advantage of these methods is that they benefit from the well developed knowledge in the generation of linear meshes [96]. The choice of the method to curve the boundary elements depends on the way $\mathcal{G}_{\text{exact}}$ is described:

2.3.1 Full Geometry Knowledge

If full knowledge of $\mathcal{G}_{\text{exact}}$ is provided (e.g. in form of a parametrisation or a CAD definition), nodal interpolations can be obtained by simply mapping the high-order interpolation nodes from the straight sided mesh to $\mathcal{G}_{\text{exact}}$. This approach is used with equidistant Lagrangian polynomials in the free mesh generator *Gmsh* [36].
As mentioned in Section 2.1.1, the positions of interpolation nodes have an effect on the quality of the geometric approximation. Elements with non equidistant nodal positions in the reference space are proposed in [85, 96, 40].
Instead of specifying the node placing on the reference element, it can also be enforced on the physical space. Therefore the node positions in the reference element have to be adapted based on the mapping to the physical element. In [96], this is done for equidistant nodes and for Fekete points. An approach for curvature dependent node placing is given in [52], where nodes are placed to divide the element in parts of equal curvature variation. In [76], the position of geometric high-order nodes is optimised with respect to the accuracy of the simulation results for the field.
As enhancement of the *Gmsh* implementation, an a-posteriori mesh optimisation routine can be applied that moves the vertices and nodes in the physical space [72]. One term of the objective function thereby controls the geometric accuracy by measuring the area between the exact and the approximating geometry. (Other terms will be described in Section 2.4.1.)
Similarly, a mesh curving approach by minimisation of a modified Fréchet distance (see Sec. 2.2.1) with respect to the nodal positions is presented in [75]. Nodes are thereby allowed to move away from the surface, resulting in non-interpolatory approximations that were found to provide better accuracies than interpolatory approximations.
While typically nodal Lagrange polynomial bases are used for the mesh description, Moxey et al. apply a modal hierarchic basis in [60, 61] that is derived by Karniadakis [50]. The choice of these bases is motivated by the possibility of varying local orders. However, both papers focus on mesh smoothing and untangling strategies and make no further comment on the curving itself or the geometric description.

2.3.2 Curve Reconstruction

If $\mathcal{G}_{\text{exact}}$ is not known, a surface reconstruction procedure has to be considered. These methods reconstruct a high-order curved surface from the linear mesh with no additional information about $\mathcal{G}_{\text{exact}}$ (no CAD file or exact parametrisation). Typically, local polynomial fittings are constructed for sets of neighbouring mesh vertices.

Recently, the mesh reconstruction software *meshCurve* was introduced by Ims et al [45, 46]. The algorithm used for the high-order reconstruction is a modification of the *Weighted Averaging of Local Fittings (WALF)* algorithm introduced by Jiao and Wang in [48]. In the original method, second order Taylor polynomials are fitted for each mesh vertex through vertices in its neighbourhood. The value of the fit for any arbitrary point on the mesh is then obtained from a weighting of its barycentric coordinates (see Figure 2.18 for a schematic in 2D). Ims et al. directly fit polynomials to the faces of the mesh in order to limit the number of averaging steps.

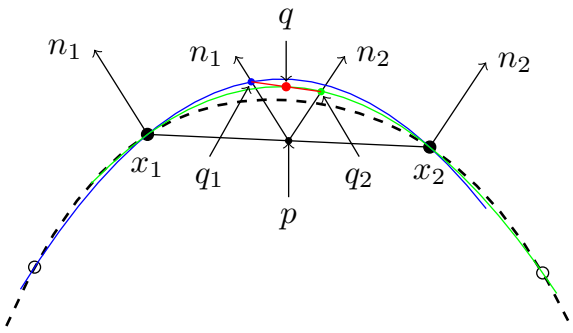


Figure 2.18: Schematic of the WALF algorithm in 2D. The black dashed line represents $\mathcal{G}_{\text{exact}}$, the blue and green line the local fittings through x_1 and x_2, respectively. q_1 and q_2 indicate the projection of point p on the two local fittings. The weighted average of q_1 and q_2 is given by point q. This figure is based on a figure in [48].

The first paper [45] seemed to indicate that only quadratic elements could be generated (compare [93]). In [46] it is clarified that geometric orders up to 6 can be achieved with *meshCurve*. Anyhow, it is also emphasised that these higher orders can suffer from oscillatory behaviour. This can not be prevented by several mitigation procedures that are included in the code. As future work, the application of orthogonal polynomial bases is suggested to stabilise the calculations. The software *meshCurve* is also equipped with mesh untangling and sharp edge detection features.

Besides the WALF algorithm, Jiao and Wang also introduce the *Continuous Moving Frames (CMF)* method [48]. It is similar to WALF, except that the fitting at a given point is not obtained from a weighted sum over the neighbouring element vertices, but rather evaluated with respect to a local coordinate system. Using this approach, CMF gets rid of the theoretically derived limitation of accuracy h^6 for WALF. However, in numerical tests, CMF does not perform significantly better than WALF.

In [91] and its subsequent paper [51], a multi-point Taylor function is used to construct elementwise quadratic extrapolations of a boundary surface from a linear mesh. The Hessian of the local extrapolation is estimated with several intermediate steps from Hessians at the edge nodes of local linear, continuous mappings. The accuracy of the quadratic approximation is shown to be dependent on the smoothness of $\mathcal{G}_{\text{exact}}$. The algorithm is only used to relocate new nodes after mesh refinement. The resulting mesh is linear again.

Spherigon patches are used in [13] as a preprocessing step (see Section 2.3.3). To evaluate a spherigon at a point $\boldsymbol{P}$ in the triangle, for each element vertex a circular arc is constructed that is orthogonal to the vertex normal. By blending the arcs, the point on the spherigon corresponding to $\boldsymbol{P}$ is found. The resulting smoothness of the resulting curve depends on the blending method [92].

2.3.3 Surface Curving from Discrete Data

Similar to surface reconstruction algorithms are methods to obtain a curved surface from discrete data. Thereby, a usually fine, pointwise or linear description of the geometry is provided. The pointwise data could come e.g. from scanning or sampling processes and are assumed to lie on the exact data. Linear interpolation of these exact data points provides the topology of the

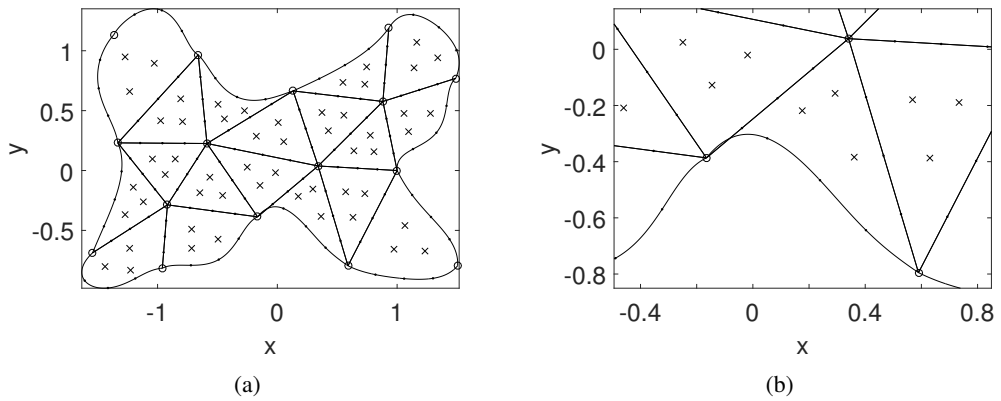


Figure 2.19: 4^{th} order mesh with a tangled element, shown (a) in total and (b) as close-up to the tangled element.

geometry and forms a first approximation of the geometry. In contrast to surface reconstruction methods, this linear mesh is finer than the final curved mesh and therefore provides additional information about the exact geometry.

Different second-order fitting methods to point-data are described and evaluated in [57]. The methods can be grouped in three different types. The first method is based on the fitting of a principal quadric in a rotated principal coordinate system. Then, extensions are considered, that improve the approximation of the surface normal direction, extend the terms of the quadric, or iterate the calculation of the surface normal and the fitted quadric. As second type, a finite difference approach is considered and an improved version with a preprocessing smoothing step. Last, quadratic patches are fitted to form a facet based approach. M^{c}Ivor and Valkenburg conclude that especially the smoothing step led to a significant improvement of the otherwise rather bad finite difference approach. The different improvement steps for the quadric fitting showed mixed results, some improving the resulting surface approximations, some not. The results with the facet based approach were similar to those of the quadric fittings, with the advantage of better computational performance.

In [13], high-order Bézier curves and patches are constructed from a coarse linear mesh and a fine linear target mesh. The positions of inner control points are determined by the minimisation of the sum over the pointwise squared distance between sampling points along the approximating curve and their normal projection on the target mesh. The control points defining the element vertices are fixed. In order to improve the results, the target mesh is enhanced by a mesh reconstruction using spherigon patches. The method is further enhanced by incrementally adding an optimisation with respect to curve bending and stretch energies ([12, 14]).

2.4 Smoothing and Untangling of Meshes

Mesh curving procedures typically curve only the boundary and leave interior edges or faces straight sided. In doing so, elements at the boundary can become tangled, meaning that the element edges (or also faces for 3D elements) are intersecting each other ([90], see Figure 2.19). Meshes with tangled elements are invalid for most numerical simulations. The elements are not

disjoint, which leads e.g. for FEM to the loss of the continuity property in the functional space [17]. Furthermore, system matrices become non-computable, or at least ill-conditioned for strongly distorted elements.
Many different methods to untangle and/or smooth meshes with invalid or highly distorted elements are given in the literature. They can be classified as optimisation based methods ([29, 32, 33, 90, 72]), methods based on elasticity analogies ([96, 60]), topological mesh modifications [53], or combinations of these techniques ([23, 71]).
Especially in the context of computational fluid dynamics (CFD), untangling methods are developed (see e.g. [1, 96, 61]). For CFD simulations, special meshes with boundary layers are needed. Boundary layer elements are thin in the normal direction of the boundary and have a high aspect ratio. Therefore, meshes for CFD simulations constitute a challenge for untangling methods [1].

2.4.1 Optimisation-based Untangling

Meshes can be smoothed by moving mesh nodes using an optimisation procedure. The objective functions are based on quality or distortion measures. Often, these measures are not defined for tangled elements, such as the condition number of the mapping from ideal to physical element for linear elements in [29]. A pre-processing step is therefore required to untangle all invalid elements.
In other cases, the measure exhibits a singularity between the domains containing the tangled and untangled elements [33]. Such a distortion measure was regularised by Escobar et al. for linear elements which removed the singularity between the domains ([24, 23]). Gargallo et al. adapted the modified measure for curved higher order elements ([32, 33]).

On the other hand, well-placed singularities provide a guarantee that untangled elements do not become invalid during optimisation. Toulorge et al. included a moving log-barrier method in their objective function to generate such a singularity (see [90]). In the paper, the objective function to untangle curved elements is built-up from two key terms. The first is an energy term which takes into account the distance between the mesh vertices on the straight-sided mesh and those on the curved mesh. By this term, main properties of the mesh such as the existence of a boundary layer can be conserved.
To enforce untangled elements, a second term is needed. This term is given as a sum of the function

$$F_l^e(x_i^e, \epsilon) = \left(\frac{B_l^e\left(x_i^e\right)}{J_0^e} - 1 \right)^2 + \log^2 \left(\frac{B_l^e\left(x_i^e\right) - \epsilon J_0^e}{J_0^e - \epsilon J_0^e} \right), \tag{2.8}$$

over all Bézier coefficients B_l at node x_i of all elements e. J_0^e denotes the Jacobian of the mapping from a straight sided reference triangle $\hat{T}$ to the straight sided triangle defined by the vertices of e. It is used to scale the Jacobian $J_{\boldsymbol{F}}$ of the mapping $\boldsymbol{F}$ from $\hat{T}$ to e. This Jacobian is a measure for the validity and shape-regularity of e: A change of sign of $J_{\boldsymbol{F}}$ within the element corresponds to tangling, values of $J_{\boldsymbol{F}}$ close to 0 indicate a strongly distorted element. The minimal and the maximal Bézier coefficients provide a lower and upper bound for $J_{\boldsymbol{F}}$ due to the convex hull property of Bézier polynomials. Thus, optimising the Bézier coefficients also optimises $J_{\boldsymbol{F}}$ and counteracts tangling.

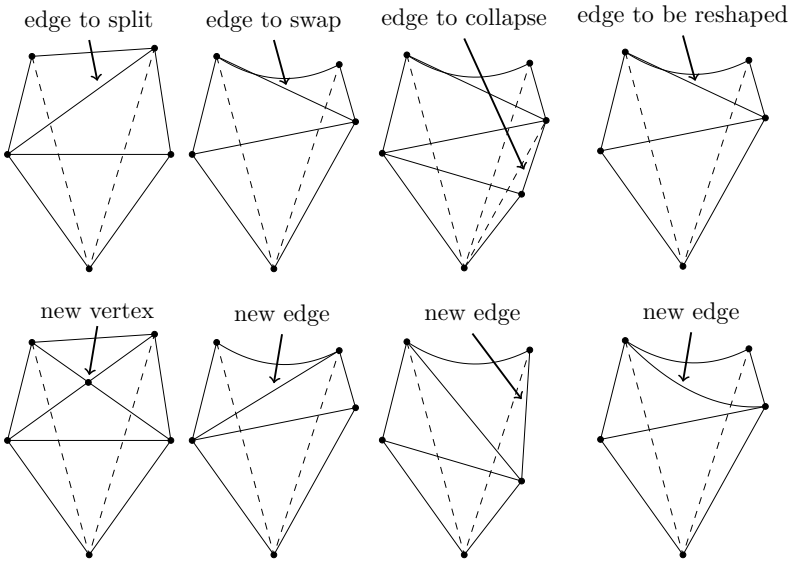
Jacobians with large values are penalised by the quadratic term. The logarithmic term is called *log barrier* and creates a singularity at ϵ. At the beginning, ϵ is set to a value just below the current minimum of the scaled Jacobian. As the mesh quality improves during optimisation and the scaled Jacobian increases, the barrier is moved accordingly. Once a minimal Jacobian target is reached, the optimisation procedure is stopped.
Since the barrier can also be set for a negative Jacobian value, untangling of invalid elements is possible. However, it is also mentioned in [90] that the two given terms alone do not necessarily work for mesh optimisation. Therefore, additional optional terms for the objective function are given, to either preserve an element's shape [90] or minimise the area between the mesh boundary $\mathcal{G}_{\text{approx}}$ and the meshed geometry $\mathcal{G}_{\text{exact}}$ [72].

2.4.2 Elasticity Analogies

For untangling based on elasticity analogies, the displacement of interior nodes is obtained by solving an elasticity equation. A linear elasticity model is considered in [96], where the stress tensor of the model is based on the tensor of node displacements. The displacement of the boundary nodes to the exact, curved geometry $\mathcal{G}_{\text{exact}}$ are enforced through Dirichlet boundary conditions and the unknowns are the displacements of the interior nodes. This way, the curvature of the boundary is propagated in the domain interior.
In [60], Moxey et al. expand the linear elasticity model by a term resulting from a thermal analogy. Depending on the considered mesh quality measure, the term represents an increased or decreased temperature, which causes the mesh elements to enlarge or tighten. Numerical results show that the additional term can improve the mesh quality.

2.4.3 Untangling by Topological Mesh Modifications

Untangling methods most often relocate the nodes without altering the mesh structure. A different approach is to make changes to the mesh topology by removing, recreating or swapping edges or faces.
In [53], an algorithm is introduced which allows to determine the optimal alteration on a set of tetrahedral elements. It relies on the representation of the mesh with Bézier curves and the resulting convex hull property. Figure 2.20 provides a selection of the applied mesh modifications: edge split, edge swap, edge collapse, and edge reshape. Note that the last case, edge reshape, is not a topological mesh modification. If the mesh is not given in a Bézier representation, criteria based on element area, angles, and edge lengths such as described in [71] can be used to detect edges for flipping, splitting or removal.
Topological mesh modifications can be combined with other smoothing or untangling techniques to increase their qualitative performance. The mesh modifications in [71] are applied together with optimisation-based untangling. The whole algorithm is not relying on a representation of $\mathcal{G}_{\text{exact}}$. Instead it uses the WALF method [48] to reconstruct a high-order description from the linear mesh.
In [23], the optimisation introduced in [24] is combined with a refinement step that changes the mesh topology. The two methods are applied in an alternating way. First the mesh is optimised, until the optimisation procedure stagnates. Then, the elements with their quality close to the



(a) Edge split (b) Edge swap (c) Edge collapse (d) Edge reshape

Figure 2.20: Selection of mesh modifications that are described in [53]. This figure is based on a figure in [53].

minimal quality are refined and the new mesh optimised again. The last two steps can be iterated a few times. Using this approach, the minimal element quality of the mesh can be increased further than with an optimisation-alone approach. For the presented numerical examples, each iteration of refinement and optimisation leads to an improvement of the minimal quality. However, note that when improving the element minimal quality, the average quality is only marginally decreased.

2.4.4 Prevention of Tangling by Iterative Node Moving

A problem with untangling methods is that they cannot really guarantee to provide an analysis-suitable mesh within reasonable runtime. In [45] for instance, the software *meshCurve* does not guarantee that the resulting mesh does not contain tangled elements. Similarly, in some cases, the untangling procedure in [90] can fail to produce a valid mesh. The success of the method partly depends on reasonable choices of parameter values. Its success rate can be increased by iterative adaption of these parameters, which also increases the runtime of the algorithm.
Another way of posing the problem is to prevent tangled elements from appearing during the curving process. This is typically done by iterative node movement, which only allows node displacements that are within the bounds of an analysis-suitable mesh.

An iterative node displacement is used in [1] to create a valid mesh of curved simplices, described by Bézier or NURBS curves. The algorithm starts form a coarse linear mesh and a piecewise Bézier or NURBS description of the curved boundary. By subdivision, a second, finer mesh is obtained which can be interpreted as the mesh of control points of the coarse mesh. A linear elasticity equation is solved for the vertex displacement of the control point mesh. Dirichlet boundary conditions are applied to enforce the boundary vertices onto the control points

of the curved boundary description. An iterative update of the vertex positions ensures that all elements remain untangled. The convex hull property of Bézier or NURBS curves guarantees that a valid mesh of control points leads to a valid curved coarse mesh.

In [49], Jiao et al. apply an iterative rescaling of node displacements that were calculated in the context of mesh smoothing for straight-sided meshes by optimisation. During the mesh update step in the smoothing algorithm, all nodes are moved at the same time, which could result in tangled elements. To prevent this, the displacements of the nodes are iteratively rescaled until they can be applied without causing any tangling.
In [71], this method is applied to prevent the formation of new tangled elements during mesh optimisation and high-order mesh reconstruction. Additionally, displacements for the mesh update are rescaled, to lie within the 1-ring neighbourhood of the original vertex. In the case of edge contraction, changes are also only applied if they do not result in tangled elements.

2.5 Conclusion of the Review

In this section, a review of geometric representations, of the evaluation of their accuracy and of methods for the enhancement was provided.
Two general ways of geometric representations were described: polynomial based and NURBS based interpolation. As it was mentioned in the Introduction, this thesis focuses on the case where no CAD or other exact geometric description is available. Therefore, the advantages of using NURBS can not be exploited and their more complicated structure makes them less favourable compared to the rather simple polynomial interpolation.
For the polynomial interpolation, again two types were proposed: nodal Lagrange polynomials and modal Lobatto polynomials. Lagrange polynomials are the standard form of polynomial interpolation in geometry representation. Lobatto polynomials on the other hand show promising properties. As modal polynomials, their representation does not rely on edge- or element-interior node positions. And the hierarchical property of the bases can be used to perform changes of interpolation orders with minimal cost. Both polynomial types will be further considered. Polynomial bases that provide smooth representations will be discarded for lack of a simple generalisation between 2D and 3D elements.
In between the presented measures of the GDE, no ranking of their significance for the acoustical simulations can be specified. Therefore several of them will be applied in Chapter 5 for the study of the dependency between the GDE and GIE, to assess their relevances. The differential geometry-based measures will not be applied due to their described disadvantages. For the GDE evaluation to compare the curving methods in Chapter 3, the area based method will be applied.
The review of mesh curving methods was divided into three categories. Most methods fall under the first category of methods that rely on the full knowledge of the exact geometry. As mentioned before and in the Introduction, this knowledge is assumed not to be given for this thesis. Curve reconstruction methods on the other side also do not fully serve the purpose, since they consider no additional knowledge of the original geometry besides the current coarse linear mesh. But they could be applied to the target mesh in a preprocessing step, improving the accuracy of the overall mesh curving process.

Such a prepocessing step was similarly applied for the last described method, which provides surface curving from discrete data. This method corresponds closely to the problem considered in this thesis. The main difference to the curving algorithms that will be proposed in Chapter 3 lies in the choice of the geometric representation. In the mentioned papers, the least squares approach shows promising results for the mesh curving, but it is not tested how appropriate the resulting curved meshes are for simulations. Therefore, the least squares approach will be adapted for the modal shape functions in this thesis. Further, curving routines with different minimisation procedures and objective functions during the curving process will be considered as well, and the resulting meshes also evaluated with acoustic simulations.
From the review of smoothing and untangling methods it became clear that this topic is already widely explored. Therefore it was decided to refrain from the development of an additional untangling method.
Considering the literature review in this chapter and the additional short reviews at the beginning of Chapters 4 and 5, the major novel elements of this thesis can be summarised as

- Development of a 2D and 3D mesh curving routine with modal Lobatto shape functions on the basis of restricted geometric knowledge.
- Comparison of nodal and modal curving methods for Helmholtz scattering simulations.
- Detection of the advantageous effect of H^1-optimised mesh curving for Helmholtz simulations.
- Quantitative study on the relation between the geometric accuracy and the geometry induced error for Helmholtz simulations.

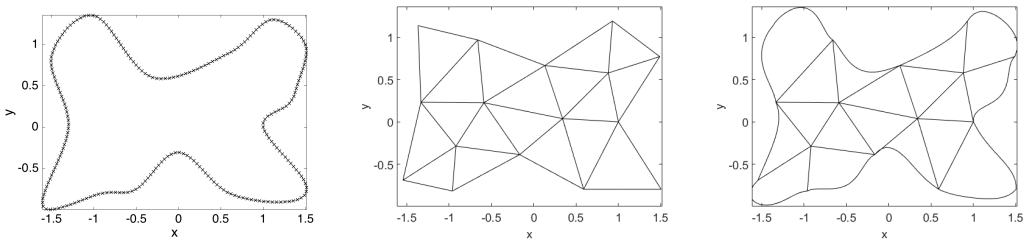
Chapter 3

Generation of 2D Meshes with Curved Boundaries

As mentioned in the Introduction and Chapter 2, the main target of this thesis is the development of a mesh curving algorithm based only on limited knowledge of the original geometry.
In this chapter, five different curving approaches are considered. Four curving algorithms are introduced that take only a fine linear mesh as input for the target curve. One is based on a nodal geometry representation using Lagrange basis polynomials. The other ones use Lobatto basis functions to obtain a modal geometry representation. An established implementation that creates high-order Lagrange elements is used as a reference. It is based on the full knowledge of the CAD representation. The curving methods are evaluated and compared with an area based GDE measure. Most parts of this Chapter were published in the conference papers [77] and [98], and the journal paper [99].
The surface meshes with curved boundaries will then be used later in Chapter 5 to analyse the influence of the geometric accuracy on the error of acoustic simulations.

3.1 Reference Lagrange Approach

The first curving algorithm that is considered is the Lagrange approach as it is implemented in the meshing software *Gmsh*. The open-source project was originally a linear mesh generator [36]. The high-order functionality was added in the later development as an a-posteriori curving after the linear meshing step. The approach is described in [35] and is based on reference element mappings with equidistant nodes on the reference element as introduced in Section 2.1.1.1. Since the curving takes place directly in the meshing environment, full knowledge of the underlying CAD-geometry is available.
For the curving of line elements or element edges, high order nodes in the physical space are first placed equidistantly along the linear edge. To map the points onto the target geometry, an orthogonal projection is used. Note that only the interior edge nodes need to be mapped as the vertex-nodes already lie on the exact geometry.
Gmsh provides high-order mesh generation up to $q = 10$. Aditionally, *Gmsh* implements optimisation algorithms that optimise the node positions [90], [72]. The objective function is thereby a weighted sum of energy terms that were described in Sections 2.3.1 and 2.4.1.



(a) (b) (c)

Figure 3.1: Schematic of (a) the fine linear Mmesh of the boundary, (b) the coarse linear Cmesh of the domain, and (c) the curved Cmesh.

Particularly, renewed mention deserves the area minimisation term

$$\mathcal{D}(\boldsymbol{x}^e) := \sum_e d_A^e(\boldsymbol{x}^e),$$

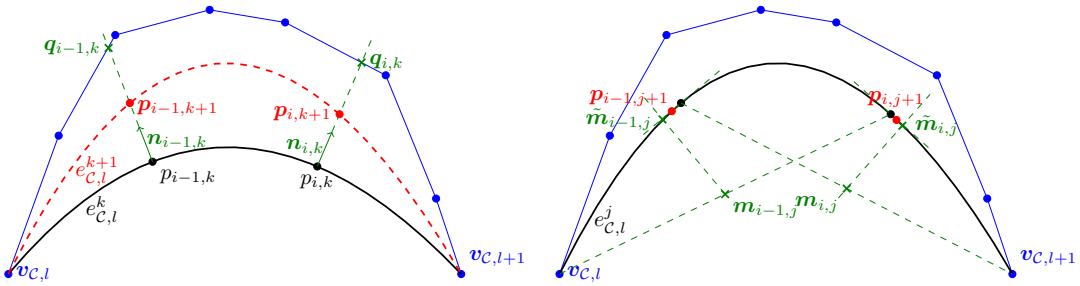
with d_A^e the area based GDE measure between an element e and the exact curve, and $\boldsymbol{x}^e$ the vector of element vertices. For the tests, optimised meshes are considered.

3.2 Cmesh-Mmesh Approaches

The reference Lagrange approach described in the last section relies on the full knowledge of the exact geometry. In applications, this knowledge is often not available. In this section, four mesh curving algorithms are presented that rely on a fine, piecewise linear approximation of the geometry as target curve. This fine target mesh is named *model mesh* or *Mmesh*. It is denoted $\mathcal{M} := (\mathcal{V}_\mathcal{M}, \mathcal{E}_\mathcal{M})$, with $n_\mathcal{M}^v$ mesh vertices $\mathcal{V}_\mathcal{M} := \left\{\boldsymbol{v}_{\mathcal{M},1}, \boldsymbol{v}_{\mathcal{M},2}, \ldots, \boldsymbol{v}_{\mathcal{M},n_\mathcal{M}^v}\right\}$, $\boldsymbol{v}_{\mathcal{M},i} \in \mathbb{R}^2$ and $n_\mathcal{M}^e$ edges $\mathcal{E}_\mathcal{M} := \left\{e_{\mathcal{M},1}, \ldots, e_{\mathcal{M},n_\mathcal{M}^e}\right\}$ that connect them.
The proposed algorithms are a-posteriori curving methods, meaning that they start off from a coarse linear mesh obtained from a linear mesh generation. The coarse mesh is curved during the algorithm and is named *computational mesh* or *Cmesh*. It is denoted $\mathcal{C}$, with $\mathcal{V}_\mathcal{C}$, $\mathcal{E}_\mathcal{C}$, $n_\mathcal{C}^e$, and $n_\mathcal{C}^v$ defined analogously to $\mathcal{V}_\mathcal{M}$, $\mathcal{E}_\mathcal{M}$, $n_\mathcal{M}^e$, and $n_\mathcal{M}^v$. For all described curving methods, it is assumed that the Cmesh vertices are a subset of the Mmesh vertices. If necessary, this can be achieved in a preprocessing step by moving the Cmesh vertices to the closest Mmesh vertices. The Cmesh-Mmesh principle is illustrated in Figure 3.1.

3.2.1 Nodal Curving Algorithm

The first introduced curving algorithm is a nodal method. The approximation of the target curve is based on the high-order nodes that are additionally placed on the line elements (compare the description of Lagrange basis polynomials in Section 2.1.1.2). The Cmesh is described by an elementwise polynomial mapping. It maps from the reference element $[-1, 1]$ to the physical space and corresponds to the mapping in the reference Lagrange approach. The basis of the mapping is formed by Lagrange polynomials with equidistant nodes in the reference space (see Section 2.1.1.2). In order to curve elementwise the Cmesh, the high-order node positions have to be found in the physical space.



(a) Curving step (b) Correction step

Figure 3.2: The figures illustrate the two different iteration steps that form the nodal curving algorithm. The blue curve represents the target curve $\mathcal{M}$. The $\mathcal{C}$ element at the beginning of the iteration step is drawn in black, whereas the red curve shows the result after the iteration step. In green, helplines and intersection points are indicated.

The algorithm starts with adding the $q-1$ high-order interpolation nodes to the linear Cmesh. On each Cmesh element, they are placed equidistantly in the physical space to form the polynomial parametrisation $\boldsymbol{g}_k : [-1,1] \to \mathbb{R}^2$ of order q. During the curving process, two operations are applied to move the high-order nodes towards the Mmesh target curve. This is done in a nested two-step iteration scheme:

1. In the outer loop, the nodes are moved along the local Cmesh normal direction to successively come closer to the Mmesh. This is done in n_{iter} iteration steps, within which each node is treated individually. In Figure 3.2(a), an example of this curving iteration step is illustrated.

   In the k^{th} iteration, the position of the i^{th} node is updated from $\boldsymbol{p}_{i,k}$ to $\boldsymbol{p}_{i,k+1}$. Therefore, the normal $\boldsymbol{n}_{i,k}$ of the parametrisation $\boldsymbol{g}_k$ at $\boldsymbol{p}_{i,k}$ is evaluated. The target point $\boldsymbol{q}_{i,k}$ for $\boldsymbol{p}_{i,k}$ on the Mmesh is evaluated as the closest intersection point between the Mmesh and $\boldsymbol{p}_{i,k}$ and the line along the normal direction. The node position is then updated as

   $$\boldsymbol{p}_{i,k+1} \quad = \boldsymbol{p}_{i,k} + \sigma_k \left(\boldsymbol{q}_{i,k} - \boldsymbol{p}_{i,k}\right), \tag{3.1}$$

   with $\sigma_k = \frac{k}{n_{\text{iter}}}$.

2. The inner iteration consists of correction steps to readjust the node spacing (see Figure 3.2(b)). The spacing in the physical space is targeted to be similiar to the node spacing in the reference space. In accordance with the equidistantly spaced nodes in the reference space, the relocation aims therefore for approximately equidistantly spaced nodes in the physical space. It is applied after each curving step.

   For node $\boldsymbol{p}_{i,j}$ in the j^{th} iteration step, the midpoint of its neighbouring points is computed as $\boldsymbol{m}_{i,j} := \frac{\boldsymbol{p}_{i-1,j}+\boldsymbol{p}_{i+1,j}}{2}$. This midpoint is then projected onto the tangent $\boldsymbol{t}_{i,j}$ in $\boldsymbol{p}_{i,j}$ to obtain the target point $\tilde{\boldsymbol{m}}_{i,j}$ for the node update. All nodes are moved at the same time by

   $$\boldsymbol{p}_{i,j+1} = \boldsymbol{p}_{i,j} + \tau \left(\tilde{\boldsymbol{m}}_{i,j} - \boldsymbol{p}_{i,j}\right), \tag{3.2}$$

   where $\tau = 0.4$ is a scaling factor that reduces the risk to overshoot. Note that τ can be chosen from $\tau \in (0,1]$. For $\tau = 1$, the point would be moved directly onto the target point,

for $\tau = 0$ it would not be moved at all.

As the correction step does generally not place the nodes exactly onto the Mmesh, one extra mesh deformation step is applied with $\sigma_{n_{\text{iter}}+1} = 1$ after the last iteration step.
To find the number of iterations that are needed for the method to converge, the number of iteration steps are increased adaptively. The distance between the nodes obtained with different iteration steps is evaluated. When the maximal distance falls below a predefined nodal tolerance of 10^{-14}, the method is declared converged. The number of iteration steps are increased for both loops simultaneously, with that of the outer loop increasing faster. If no convergence is observed until the number of iteration steps of the outer loop reaches the upper limit of 200 iterations, the algorithm breaks down.

3.2.2 Modal Curving Algorithms

Besides the nodal approaches, modal curving methods are considered. They rely on a description of the curved mesh with the hierarchic modal Lobatto polynomial basis which was introduced in 2.1.1.4. As with Lagrange basis polynomials, the curves are generally described by a linear combination $\hat{\boldsymbol{f}} = \sum_{i=1}^{n_q} \boldsymbol{\alpha_i} l_i$, with l_i the i^{th} Lobatto polynomial. The main difference to the nodal approaches is that there are no high-order nodes to define the geometry. Instead, the coefficients $\boldsymbol{\alpha_i}$ have to be determined in the curving process.
From the polynomial approximation $\hat{\boldsymbol{f}}(\boldsymbol{\xi})$, the linear interpolation with the first-order shape functions is known by the Cmesh vertex positions $\boldsymbol{p}_1 = (p_{1,x}, p_{1,y})$ and $\boldsymbol{p}_2 = (p_{2,x}, p_{2,y})$. Thus, only the interpolation with the higher-order shape functions (which will be called the *polynomial deflection*) and the corresponding $\boldsymbol{\alpha}_j$ with $j = 2, \dots, q$ have to be determined in the curving process.

$$\hat{\boldsymbol{f}}(\xi) = \underbrace{\frac{1}{2}(-\xi+1)\boldsymbol{p}_1 + \frac{1}{2}(\xi+1)\boldsymbol{p}_2}_{\text{linear interpolation}} + \underbrace{\sum_{i=2}^{q} \boldsymbol{\alpha}_i \boldsymbol{l}_i(\xi)}_{\text{polynomial deflection}} \qquad \text{for } \xi \in [-1,1]\,.$$

Hereafter, three a-posteriori methods are described to obtain the $\boldsymbol{\alpha}_i$ coefficients. The first is a least squares approach. There, only the Mmesh vertices are considered to describe the target curve. These nodes lie on the exact geometry. The other two methods are based on an optimisation with respect to the H^1-seminorm. This allows to take full advantage of the δ-property (2.5) of the Lobatto shape functions and bypasses matrix inversions. The optimisation will be considered in two different function spaces.

3.2.2.1 Least Squares Modal Curving

The first curving method follows the standard least squares approach by minimising a squared error. Therefore, the error is evaluated as the squared distance between the Mmesh vertices and corresponding points on the Cmesh. As the measure is pointwise, no interpolation is necessary between the Mmesh vertices, and only geometrically exact information is considered. In [13], a similar approach is described that applies Bézier curves for the geometric approximation.

A single Cmesh element is curved by approximating the n target Mmesh vertices (x_i, y_i) with the polynomial $\hat{\boldsymbol{f}} = \sum_{j=0}^{q} \boldsymbol{\alpha}_j \boldsymbol{l}_j(\xi)$. The Mmesh arc is parametrised based on the element length with

$$\xi_i = -1 + 2\frac{\sum_{j=1}^{i-1} h_j}{\sum_{j=1}^{n-1} h_j} \in [-1, 1] \tag{3.3}$$

corresponding to (x_i, y_i) and h_j the length of the j-th Mmesh element on the arc. This parametrisation allows to find the corresponding points on the Cmesh for each Mmesh vertex. The minimisation that has to be fulfilled by the coefficients $\boldsymbol{\alpha}_j$ is then

$$min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \|(x_i, y_i) - \hat{\boldsymbol{f}}(\boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_q; \xi_i)\|_2^2.$$

The minimisation is transformed into a system of linear equations by taking the derivatives with respect to the individual α_js and setting the resulting equation to 0. Both the x- and y coordinates are described independently. For each coordinate this results in an independent linear system, with $\boldsymbol{\alpha}_i = (\alpha_{i,x}, \alpha_{i,y})$. The derivation is shown for the x-coordinate. The subindex x is dropped for readability.

$$\sum_{j=2}^{q} \alpha_j \underbrace{\sum_{i=1}^{n} l_k(\xi_i) l_j(\xi_i)}_{=:a_{kj}} = \underbrace{\sum_{i=1}^{n} x_i l_k(\xi_i) - \sum_{j=0}^{1} \alpha_j \sum_{i=1}^{n} l_k(\xi_i) l_j(\xi_i)}_{=:b_k}$$

$$\Leftrightarrow \boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{b},$$

with

$$\boldsymbol{A} = \begin{pmatrix} a_{2,2} & \cdots & a_{2,q} \\ \vdots & \ddots & \vdots \\ a_{q,2} & \cdots & a_{q,q} \end{pmatrix}, \quad \boldsymbol{\alpha} = \begin{pmatrix} \alpha_2 \\ \vdots \\ \alpha_q \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} b_2 \\ \vdots \\ b_q \end{pmatrix},$$

where $\boldsymbol{A}$ is a generally fully-populated matrix. The system for the y-coordinate follows analogously.

3.2.2.2 H^1 Modal Curving

A second modal based curving method is presented that is based on an optimisation in the H^1-seminorm. Taking advantage of the δ-property (2.5) of the Lobatto shape functions, an explicit formula can be derived for the coefficients $\boldsymbol{\alpha_i}$. The general derivation is described in [82] as *projection based interpolation*. The optimisation is considered for each coordinate independently and accordingly, a set of α_i-values is obtained for each coordinate. Here, the derivation is outlined for the x-coordinate α_i-values that approximate the target f_x by $\hat{f}_x$. For simplicity, the x-index is dropped for the α_i. The y-coordinate α_i-values are found analogously.

The difference between the target f_x and its approximation $\hat{f}_x$ shall be minimised in the

H^1-seminorm. Hence, the α_i have to fulfil the optimisation problem

$$min_{\boldsymbol{\alpha}}|f_x - \hat{f}_x|_{H^1} = min_{\boldsymbol{\alpha}} \int_{-1}^{1} (f_x' - \hat{f}_x{}')^2 \mathrm{d}\xi.$$

Note that this optimisation minimises the deviation of the parametrisations' derivatives. These are directly linked to the curves tangents and normals. The minimisation in the H^1-seminorm therefore corresponds to an optimisation of the normal approximation.

As described above, the linear interpolation of the Cmesh vertices is already defined by the known α_0 and α_1. Thus, the target curve is split up in the known linear part and the remaining deflection d

$$f_x = \underbrace{\alpha_0 l_0 + \alpha_1 l_1}_{\text{linear part}} + d$$

and the optimisation reduces to

$$min_{\boldsymbol{\alpha}} \int_{-1}^{1} (d' - \sum_{i=2}^{q} \alpha_i l_i'(\xi))^2 \mathrm{d}\xi.$$

For the α_k that solve the minimisation it holds that

$$\frac{\partial}{\partial \alpha_k} \int_{-1}^{1} (d' - \sum_{i=2}^{q} \alpha_i l_i'(\xi))^2 \mathrm{d}\xi = 0.$$

This can be reformulated to obtain an explicit formula for the coefficients

$$\begin{aligned}
\frac{\partial}{\partial \alpha_k} \int_{-1}^{1} (d' - \sum_{i=2}^{q} \alpha_i l_i'(\xi))^2 \mathrm{d}\xi &= 2 \int_{-1}^{1} (d' - \sum_{i=2}^{q} \alpha_i l_i'(\xi)) l_k'(\xi) \mathrm{d}\xi \\
&\overset{(2.5)}{=} 2 \int_{-1}^{1} d' l_k' \mathrm{d}\xi - 2\alpha_k \\
\Rightarrow \alpha_k &= \int_{-1}^{1} d' l_k' \mathrm{d}\xi.
\end{aligned}$$

Note that, as the target curve is the piecewise linear curve connecting the Mmesh vertices, d is generally non-polynomial. Further, its formula depends on the parametrisation of the Mmesh arc over the Cmesh element. Here, a parametrisation based on the physical lengths of the Mmesh elements is applied. The resulting formulas for the deflection are

$$\begin{aligned}
d_i(\xi) = &\frac{\sum_{j=1}^{n-1} h_j}{2h_i}(x_{i+1} - x_i) \left(\xi - (-1 + \frac{2}{\sum_{j=1}^{n-1} h_j} \sum_{j=1}^{i-1} h_j) \right) + x_i \\
&- \underbrace{\left(\frac{1}{2}(-\xi + 1) p_{1,x} + \frac{1}{2}(\xi + 1) p_{2,x} \right)}_{\text{linear part}} \\
d_i'(\xi) = &\frac{\sum_{j=1}^{n-1} h_j}{2h_i}(x_{i+1} - x_i) + \frac{1}{2} p_{1,x} - \frac{1}{2} p_{2,x},
\end{aligned}$$

with $d = d_i$ if $\xi \in [\xi_i, \xi_{i+1}]$, n the number of Mmesh vertices on the Mmesh arc, h_j the lengths

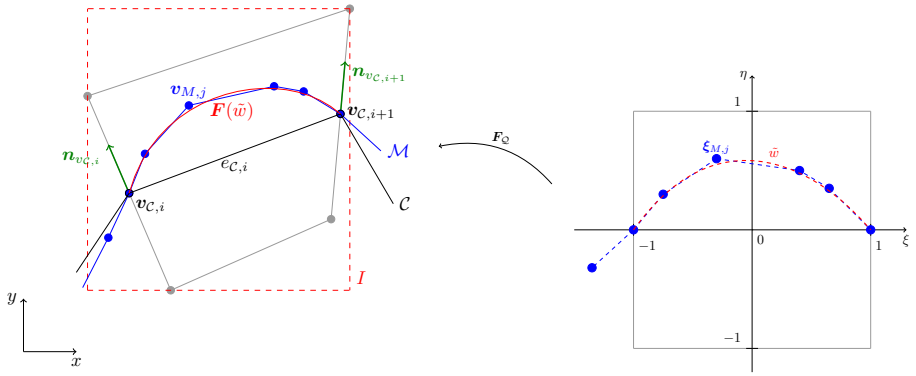


Figure 3.3: Illustration of the reference-based modal method with the physical space on the left and the reference space on the right. The $\mathcal{M}$ drawn in blue, $\mathcal{C}$ with edge e_i in black, the quadrangle spanned by e_i in grey, and the bounding interval I in red. In the reference space, the blue curve represents the local function w and the red one the polynomial interpolation $\tilde{w}$. The reference quadrangle is indicated in grey.

of the j^{th} Mmesh element on the Mmesh arc, and ξ_i as defined in (3.3). The resulting formula for the α_k is

$$\alpha_k \quad = \quad \sum_{i=1}^{n-1} \left(\int_{\xi_i}^{\xi_{i+1}} \frac{\sum_{j=1}^{n-1} h_j}{2h_i} (x_{i+1} - x_i) l_k'(\xi) \mathrm{d}\xi \right) + \left(\tfrac{1}{2} p_{1,x} - \tfrac{1}{2} p_{2,x}, \right) \int_{-1}^{1} l_k'(\xi) \mathrm{d}\xi.$$

This algorithm will be referred to as H^1 *modal curving*.

3.2.2.3 Reference-based Modal Curving Algorithm

The third considered method will be called *reference based modal curving*. As in the H^1 curving approach, it is based on an optimisation in the H^1-seminorm. For this method, the Mmesh vertices are mapped onto a reference space. Their linear interpolation is then approximated. In the (ξ, η) - reference space, the linear Cmesh element is mapped onto the $[-1, 1]$ interval on the ξ-axis. The advantage of this method is that only one set of α_i-values for the η-coordinate has to be calculated. The final curve is obtained by mapping the polynomial interpolation back into the physical space. The method is illustrated in Figure 3.3.

The main difference to the previous method lies in the mapping $\boldsymbol{F}_{\mathcal{Q}}(\xi, \eta)$ from the reference to the physical space. It maps the standard $[(-1, -1), (1, 1)]$ reference quadrangle to a straight sided quadrangle in the physical space that is constructed around the Cmesh element. Therefore, the Cmesh vertices are extruded in positive and negative vertex normal directions. The extrusion length is predefined to ensure that the quadrangles are untangled, dinstinct and that the whole Mmesh is covered by their union.

Those Mmesh vertices that lie in the bounding interval I of the quadrangle are inversely mapped onto the reference space. As generally no closed-form expression of $\boldsymbol{F}_{\mathcal{Q}}^{-1}$ is available, the corresponding points in the reference space are obtained by minimising the objective function $g(\xi, \eta) := |\boldsymbol{x} - \boldsymbol{F}_{\mathcal{Q}}(\xi, \eta)|$.

The target curve in the reference domain, defined as the linear interpolation of the mapped Mmesh vertices, is denoted by w. Note that this linear interpolation only approximates the inverse image of $\mathcal{M}$, as $\boldsymbol{F}_{\mathcal{Q}}$ is generally not linear.

Analogously to the H^1 modal curving, the α_i-values to approximate the target curve are defined by an optimisation in the H^1-seminorm

$$\boldsymbol{\alpha} := \operatorname{argmin}_{\boldsymbol{\alpha}} |w - \tilde{w}|_{H^1} = \operatorname{argmin}_{\boldsymbol{\alpha}} \int_{-1}^{1} (w - \tilde{w})' (w - \tilde{w})' \, \mathrm{d}\xi,$$

Again, the linear interpolation is predefined by $w(-1) = \alpha_0 = w(1) = \alpha_1 = 0$ and a closed-form expression of the remaining $\alpha_2 \dots \alpha_q$ is found as

$$\alpha_i = \int_{-1}^{1} w'(\xi) l_i(\xi) \, \mathrm{d}\xi, \quad \text{for } i = 2, \dots, q.$$

3.3 Comparison of the Methods

In this Section, the introduced curving methods are compared. They are applied to two test geometries. Additionally, the applicability and the resulting accuracy in the geometric approximation are analysed.

3.3.1 Circle Test Case

The first test geometry is the unit circle. It will be also applied later for simulation tests (see Section 4.4.1).

3.3.1.1 Geometry and Meshes

For the reference Lagrange approach, the geometry is defined by four quadrants using the *Gmsh*-internal command *Circle*. A mesh with evenly sized elements is created and then curved. For the nodal and the modal curving approaches, the Mmesh and Cmesh are generated the same way. Before the curving, the assumption $\mathcal{V}_\mathcal{C} \subset \mathcal{V}_\mathcal{M}$ has to be fulfilled. Therefore, each Cmesh vertex is snapped onto the closest Mmesh vertex, where the distance is measured in Euclidean distance. Note that this choice, instead of using the geodesic distance, could lead to an invalid Cmesh for thin geometries. In this thesis, the Mmesh is assumed to appropriately describe the geometry, including a high enough refinement to avoid such problems. This is achieved by a refinement the same size as the geometry thickness, as one half of the Mmesh element size is the maximum distance between a Cmesh vertex and its closest Mmesh vertex.

3.3.1.2 GDE Measure for the Comparison

The accuracy of the curving is evaluated using a measure based on the area A between the exact circle and its approximating curve. A similar measure was applied in [72]. The main advantage of area-based measures is that they are independent of the curve parametrisation, as opposed to the described measures based on the curve normal or curvatures (see Section 2.2.4). Further, in [72] they were found to show some correlation with the Hausdorff distance and are much easier to compute than the latter.
For this test case, the parametrisation of the exact boundary and its inverse are known. This knowledge can be used to calculate the area directly. First, the inverse parametrisation of the

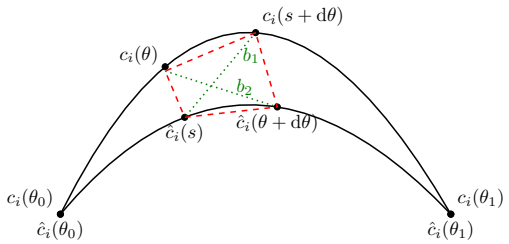


Figure 3.4: Sketch of the ODE-based calculation of the area between two parametrised curves.

angle based parametrisation $c_i(\theta)$ of the circle is used to find the parameter interval $[\theta_1, \theta_2]$ that corresponds to the i^{th} Cmesh element. The parametrisation $\hat{c}_i(\theta)$ of its approximation is then constructed over the same reference interval. Following the area formula $A_{\text{quad}} = \frac{1}{2}|b_1 \times b_2|$ of a quadrilateral with diagonals b_1 and b_2, the increase of the area between the two curves when travelling along the parametrisation from θ to $\theta + \mathrm{d}\theta$ can be evaluated as

$$A_i(s+\mathrm{d}s) - A_i(s) = \frac{1}{2}\left|(c_i(s+\mathrm{d}s) - \hat{c}_i(s)) \times (c_i(s) - \hat{c}_i(s+\mathrm{d}s))\right|$$

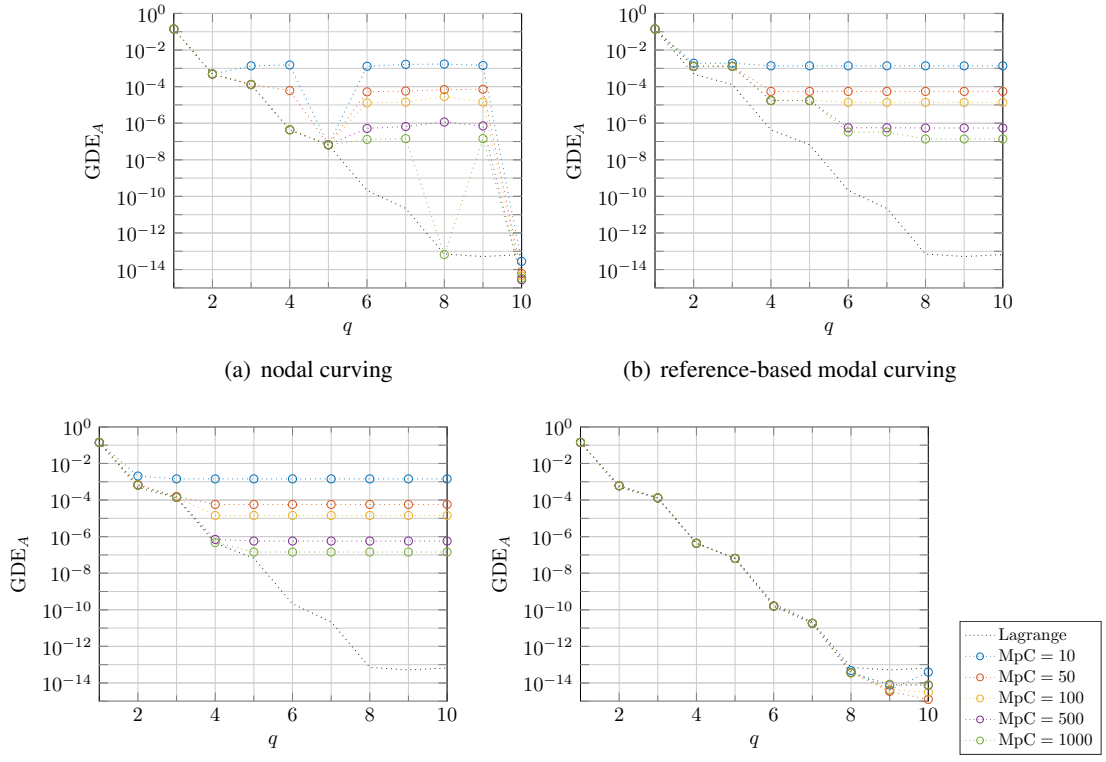
for a small step $\mathrm{d}\theta$ (see Fig. 3.4).
After linear Taylor series expansions of c_i and $\hat{c}_i$, the formula can be rearranged to the ordinary differential equation

$$\frac{\mathrm{d}A_i}{\mathrm{d}s}(s) = \frac{1}{2}\left|(\hat{c}_i(s) - c_i(s)) \times \left(\hat{c}_i'(s) + c_i'(s)\right)\right|. \tag{3.4}$$

By solving (3.4) with the initial value $A_e(s_{0,i}) = 0$, the area between the two curves is obtained. This approach bypasses the approximation of both curves by a polygon as it is done in [72].
For the total area, the elementwise areas A_i are summed up as $\text{GDE}_A := A = \sum_i A_i$.

3.3.1.3 Results

Figure 3.5 shows q-convergence results for nodally and for modally curved elements with varying numbers of Mmesh elements per Cmesh element. Due to the construction of $\mathcal{C}$ and $\mathcal{M}$, this number is constant and can be calculated as $\text{MpC} = n^e_{\mathcal{M}}/n^e_{\mathcal{C}}$. The GDE_A is plotted for the case $n^e_{\mathcal{C}} = 12$. The results for the Lagrange elements are included as reference.
For all five methods, the q-convergence plots show an odd-even stepwise behaviour. This is linked to the specific circle geometry, as it is not observed in later test cases (see Section 3.3.2). The first intuition is that the behaviour is linked to the symmetry of the circle arcs and the symmetry of the shape functions. For the reference-based modal curving this is confirmed by the fact that the α values corresponding to odd order shape functions are of order 10^{-13}. Lobatto basis polynomials can be split in those that are of even order and axially symmetric in $\xi = 0$ ($l_{2k}(-\xi) = l_{2k}(\xi)$), and those of odd orders that are anti-symmetric with respect to the origin ($l_{2k+1}(-\xi) = -l_{2k+1}(\xi)$). In the reference space, the target curve is defined in one coordinate only and over the same parameter interval as the Lobatto shape functions. Due to the circle's symmetry, it is further axially symmetric in $\xi = 0$, like the Lobatto polynomials of even order. Therefore, no significant contribution is observed from the Lobatto basis polynomials of odd



(a) nodal curving

(b) reference-based modal curving

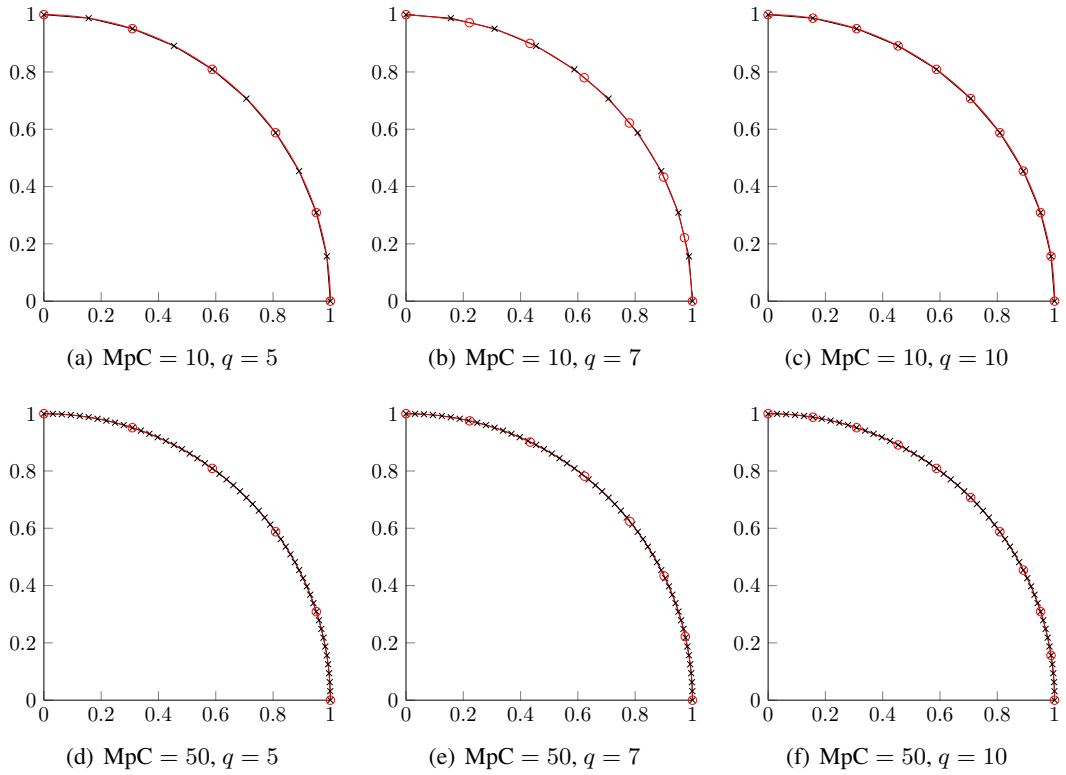
(c) H^1 modal curving

(d) least squares modal curving

Figure 3.5: Plot of the relative area between the exact circle and the approximation curved with (a) the nodal curving approach, (b) the reference-based modal approach, (c) the H^1 modal curving or (d) the least squares approach, for $n_C^e = 12$ and different ratios MpC.

MpC	10	50	100	500	1000
area	$1.4353 \cdot 10^{-3}$	$5.7419 \cdot 10^{-5}$	$1.4355 \cdot 10^{-5}$	$5.7419 \cdot 10^{-7}$	$1.4359 \cdot 10^{-7}$

Table 3.1: Area between the Mmesh and the exact circle.



(a) MpC $= 10$, $q = 5$ (b) MpC $= 10$, $q = 7$ (c) MpC $= 10$, $q = 10$

(d) MpC $= 50$, $q = 5$ (e) MpC $= 50$, $q = 7$ (f) MpC $= 50$, $q = 10$

Figure 3.6: Plots showing the Mmesh vertices in black and the high-order nodes of the nodally curved Cmesh in red. For $q = 5$ and $q = 10$, the Cmesh nodes coincide with geometrically exact Mmesh vertices. This is not the case for $q = 7$.

order. For all other methods, the stepwise behaviour is less strong and an evaluation of the contributions of odd and even shape functions could not confirm that only the even shape functions contribute to the approximation of the circle arc.

For the results of the nodal curving, two combined effects are observed. The curves show a stagnating behaviour. The stagnation level decreases as MpC increases. It corresponds to the area between the Mmesh and the exact circle (see Table 3.1). Therefore, the stagnation is explained by the inaccuracy of $\mathcal{M}$ with respect to the exact circle.

For some orders q, the results deviate from this accuracy bound and fall together with the Lagrange reference. More accurately, this happens for orders that are divisors of the MpC. In these cases, the higher order nodes of the nodal approach fall onto $\mathcal{M}$ vertices. Therefore, they are exact and the resulting curve corresponds to the standard Lagrange curve (see Fig. 3.6). Note that this behaviour is due to the symmetry of the circle arc over each Cmesh element. If the Mmesh is further refined, the error stagnation level becomes less significant. Then it can be observed that also for non-divisor numbers of nodes, where the nodes do not fall onto Mmesh vertices, the nodal approach has about the same accuracy as the reference approach (see Figure 3.7 for 4 Cmesh elements, $\mathrm{MpC} = 5000$, $q = 7$).

For the reference based modal curving approach, the q-convergence curves stagnate. The onset of

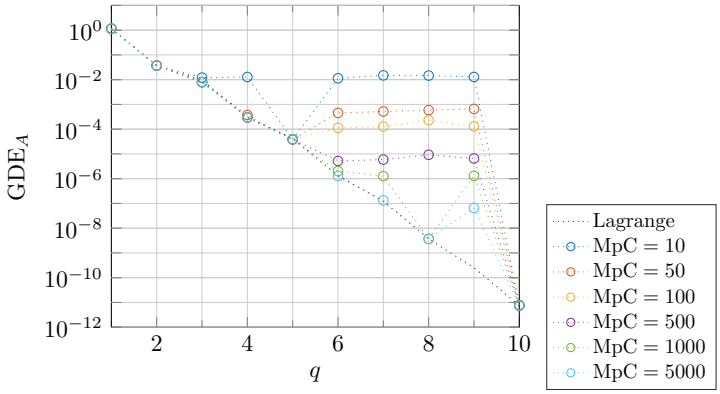


Figure 3.7: GDE evaluation for the unit circle, approximated by the nodal approach with 4 elements. The high Mmesh refinement leads to a relatievly low stagnation error. The graph shows that outside the stagnation regime, the nodal approach leads to about the same GDE as the reference approach

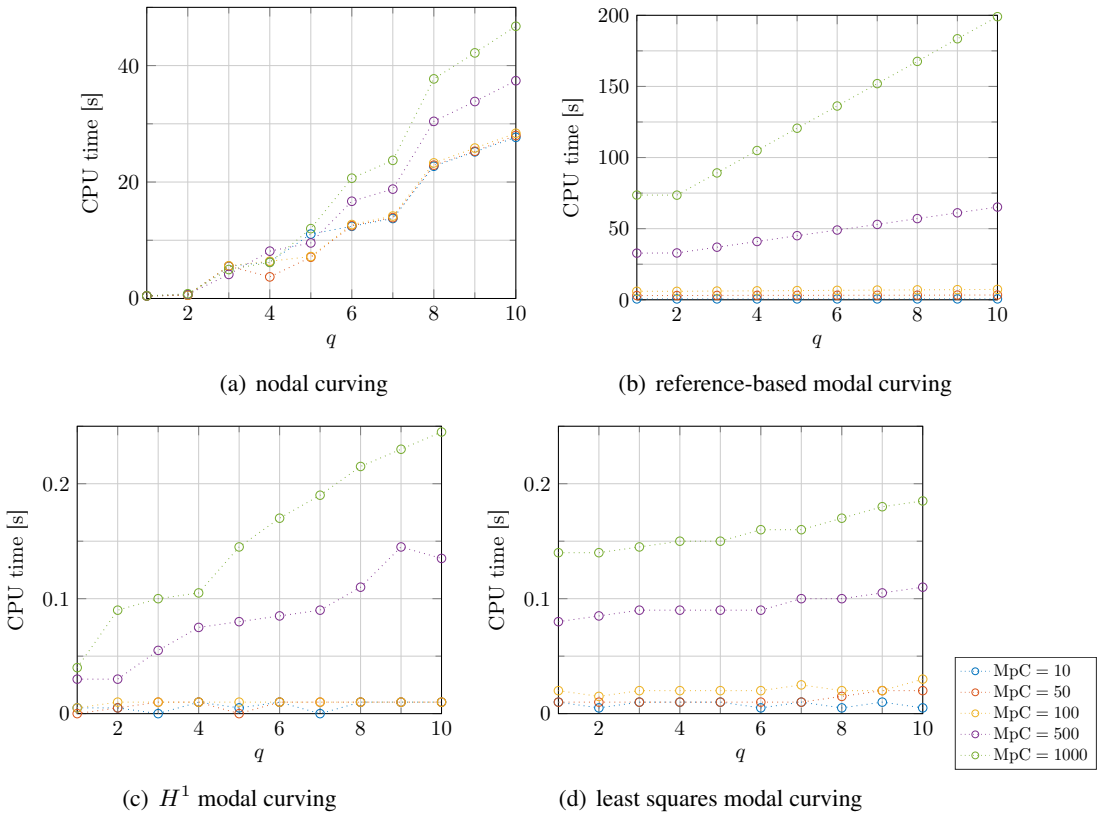
the stagnation depends on the Mmesh refinement and the corresponding Mmesh accuracy. As the polynomial approximation of the target curves improves with increasing q, the inaccuracy of $\mathcal{M}$ becomes the dominating source of error. The coarser the Mmesh, the higher its inaccuracy and the earlier the resulting error stagnation level is reached.

Further, a convergence gap can be observed between the results of the reference-based modal curving and those of the reference Lagrange approach. This can be explained by the fact that for the reference-based modal approach, the optimisation for the curving takes place in a restricted function space. The mapping from the reference space to the physical reduces the number of α_i values to determine in the curving step, but it results in a non-optimal reduction of the degrees of freedom in the optimisation.

The optimisation in the H^1 modal curving approach on the other hand is run in the full polynomial function space. Its results do not suffer from the described convergence gap (see Figure 3.5(c)) and are therefore significantly better than those of the reference based modal curving. As for the nodal curving and the reference based modal approach, the error curves stagnate due to the inaccuracy of the Mmesh. But in the region of convergence, they show the same behaviour as the curves of the results with the *Gmsh* reference.

Figure 3.5(d) shows the results of the least squares method. For this approach, the optimisation is performed in the physical space and only the geometrically exact Mmesh vertices are considered. Therefore, the convergence curves follow those of the reference. Further, no stagnation is observed as in the other three methods because the linear interpolation between the Mmesh vertices is ignored. The method provides accurate approximations already with a relatively coarse Mmesh, where the curves stagnate at error levels that are close to machine precision. In this region, higher errors for finer $\mathcal{M}$ seem to be caused by accumulation of the numerical errors.

To evaluate the runtime of the four methods, the curving of the circle with 12 Cmesh elements is considered (see Figure 3.8). For each setup, the curving is run ten times and the median of the measured CPU times taken to smoothen the resulting curves. Note the different scales of the time axis. All four methods are implemented in Matlab without focus on runtime optimisation. Therefore, only general, significant effects are compared. As expected, the run times increase



(a) nodal curving

(b) reference-based modal curving

(c) H^1 modal curving

(d) least squares modal curving

Figure 3.8: Plot of the CPU time for the curving with (a) the nodal curving approach, (b) the reference-based modal approach, (c) the H^1 modal curving or (d) the least squares approach, for $n_{\mathcal{C}}^e = 12$ and different ratios MpC.

with the refinement of the Mmesh and for higher curving orders. The nodal and the reference based modal approach both have a significantly longer runtime than the H^1 modal and the least squares approach. For the nodal approach, this originates from the adaptivity of the iterative process. A better choice of the parameters of the adaptivity loop might help to improve this. In the case of the reference based modal approach, a big overhead of calculations to set up the reference space and to transform the data points into the latter leads to the long run times. The CPU times of the H^1 modal curving and the least squares approach are of similar magnitude. For lower curving orders, the H^1 modal curving seems advantageous. With increasing q, this approach takes longer, while the runtime with the least squares approach stays nearly constant and the latter becomes favourable. In general, the run times of these two methods can be neglected compared to the simulation runtimes, especially as the curving is run only once and the resulting mesh can be applied for several simulations.

3.3.2 Distorted Ellipse

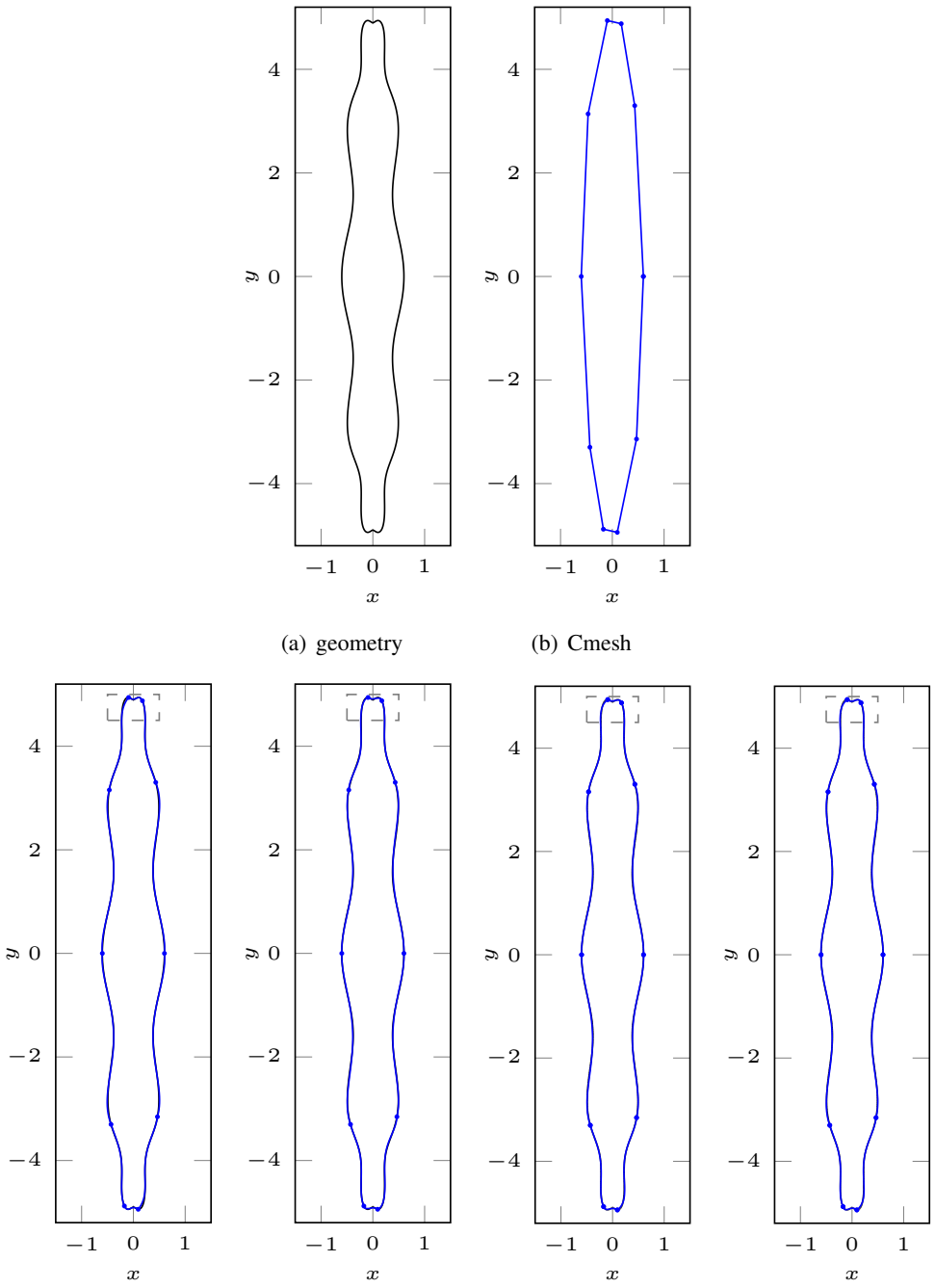
With this second test case, the curving approaches are compared for a more complicated geometry. The test geometry is based on an ellipse with a high aspect ratio. Additional bumps along the curve are introduced by a cosine term

$$\begin{pmatrix} x(\theta) \\ y(\theta) \end{pmatrix} = \begin{pmatrix} (r_x + h_b \cos(\theta n_b)) \cos(\theta) \\ (r_y + h_b \cos(\theta n_b)) \sin(\theta) \end{pmatrix} \quad \text{for } \theta \in [0, 2\pi] \,, \tag{3.5}$$

with radii $r_x = 0.5$, $r_y = 5$, $n_b = 10$ the number of bumps and $h_b = 0.1$ their height. In Figure 3.9(a), a plot of the geometry is shown.
Mildly irregular boundary meshes are considered, as they also might occur in applications. The meshes are generated in three steps. To obtain a mesh with N^v vertices, $n_v := {}^{(N^v+2)}/_2$ equidistant nodes $\tilde{\theta}_i = \frac{(i-1)\pi}{n_v-1} \in [0, \pi]$ are defined in the reference space. In a second step, these nodes are perturbed in the form of $\theta_i = \tilde{\theta}_i + \gamma_i$, where γ_i are uniformly distributed pseudo random values in the interval $[-\frac{1}{3n_v}, \frac{1}{3n_v}]$ and $\gamma_0 = \gamma_{n_v} = 0$. The pseudo random numbers are generated in *Matlab R2015b* using the *rand* command with seed 28. Applying the curve parametrisation (3.5) to the θ_i constructs the mesh vertices for the upper half of the geometry. The other half is created by mapping $\theta_i + \pi$ onto the physical space. Thereby the choice of $\gamma_0 = \gamma_{n_v} = 0$ ensures that a closed curve is obtained. As a last step before the curving, the Cmesh vertices are snapped onto the Mmesh vertices, as it was described for the previous test case. An example Cmesh is plotted in Figure 3.9(b).
Figures 3.9 and 3.10 provide an example of the initial $\mathcal{C}$ and the final curved mesh with all methods for $n^e_{\mathcal{C}} = 10$, $n^e_{\mathcal{M}} = 998$, and $q = 4$,

3.3.2.1 GDE Measure for the Comparison

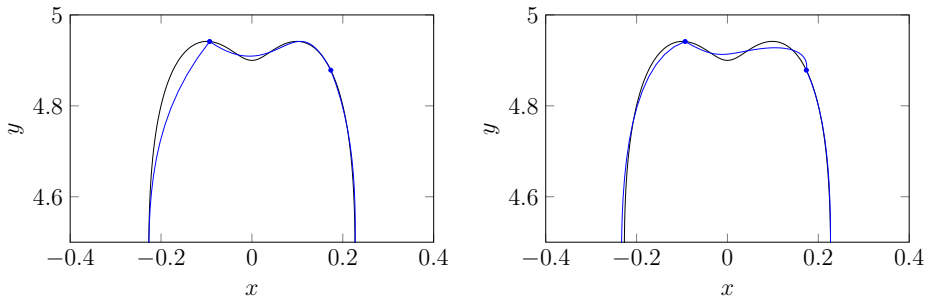
As for the previous test case, an area based GDE is evaluated. Here, the area between the target curve $\mathcal{M}$ and its approximation is calculated. This is motivated by the fact that no explicit inverse of the parametrisation is available, as it was used in the GDE calculation for the circle test case. A similar issue would be encountered for real life geometries, where we assume that only the



(a) geometry (b) Cmesh

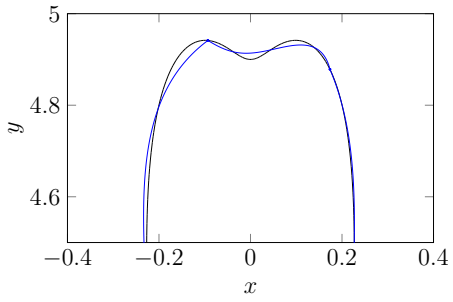
(c) nodal curving result (d) ref. based modal curving result (e) H^1 modal curving result (f) least squares curving result

Figure 3.9: (a) shows the distorted ellipse defined by (3.5) and (b) the Cmesh with $n^e_{\mathcal{C}} = 10$. The 4^{th}-order curves resulting with $n^e_{\mathcal{M}} = 998$ from (c) the nodal curving approach, (d) the reference based modal curving approach, (e) the H^1 modal curving approach, and (f) the least squares curving approach are plotted in blue against the exact geometry in black. The grey dashed boxes indicate the zoom-in area that is shown in Figure 3.10

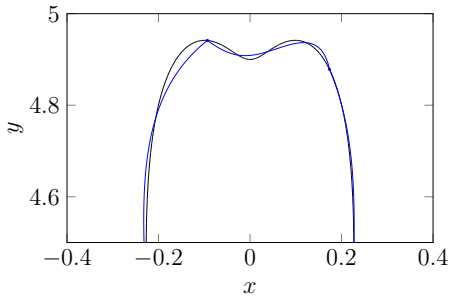


(a) zoom on nodal curving result

(b) zoom on ref. based modal curving result



(c) zoom on H^1 modal curving result



(d) zoom on least squares curving result

Figure 3.10: Close-up plots of the zoom-in regions indicated in Figure 3.9.

Mmesh is provided. Further, only the four Cmesh-Mmesh approaches are compared among each other. The area is divided into small quadrangles $\mathtt{Q}_i = \left(\mathtt{q}_{i,1}, \mathtt{q}_{i,2}, \mathtt{q}_{i,3}, \mathtt{q}_{i,4}\right)$, for which the area can be calculated with the following formula for convex quadrangles:

$$A_{\mathtt{Q}_i} = \frac{1}{2} \left| \left(\mathtt{q}_{i,2} - \mathtt{q}_{i,4}\right) \times \left(\mathtt{q}_{i,1} - \mathtt{q}_{i,3}\right) \right| .$$

The number of non-convex quadrangles is reduced by ensuring that all Mmesh vertices and all intersections between the Cmesh and the Mmesh coincide with a quadrangle vertex. The intersections are found with an adapted version of the *interX* function from [63]. Further, the number of quadrangles is adaptively changed until a predefined accuracy of 10^{-9} is reached in the area calculation. A relative area measure is calculated, similar to the measure suggested in [72]. Therefore, the sum over all quadrangles' areas is divided by the length $l_{\mathcal{M}}$ of $\mathcal{M}$:

$$\mathrm{GDE}_{A_{\mathtt{Q}}} := \frac{\sum_i A_{\mathtt{Q}_i}}{l_{\mathcal{M}}}.$$

3.3.2.2 Results

Fixed number of Cmesh elements

First, the size of the Cmesh is fixed with $n_{\mathcal{C}}^e = 20$ and the target Mmesh is varied with

$$n_{\mathcal{M}}^e \in \{98, 198, 298, \ldots, 998, 1998, 2998, \ldots, 9998\} .$$

This corresponds to the study for the circle case in Section 3.3.1. Since the Cmesh and Mmesh are not generated with equisized elements, the ratio $\frac{n_{\mathcal{M}}^e}{n_{\mathcal{C}}^e}$ does not represent anymore MpC, the

number of Mmesh elements for each Cmesh element, but its mean $\widetilde{\mathrm{MpC}}$ over all Cmesh elements. Figure 3.11 shows the corresponding q-convergence curves. Note that for the nodal approach no data points are available for $q > 8$. For these cases, the curving algorithm failed to converge within the iteration boundaries mentioned above.

The general behaviour of the results correspond to those of the circle test case, with all four methods showing the same trends.

For lower curving orders q, the curves converge, but then stagnate as q increases, depending on the Mmesh refinement. The least squares does not provide any significant advantage. Similarly, the results with all four methods show approximately the same convergence rate and no convergence gap is observed for the reference based method. It actually provides the best results for some $\mathcal{M}$ and q combinations.

The less regular geometry leads to a loss of the stepwise behaviour. Odd and even order Lobatto shape functions seem therefore to contribute to a similar extend. The only significant outlier is the case of $q = 7$ for the three methods that curve the Cmesh directly in the physical space. There, the polynomial of order $q = 6$ seems to describe the distorted particularly well, while the shape functions of order $q = 7$ do not improve the approximation.

With the coarse $\mathcal{M}$, one disadvantage of the least squares approach becomes visible. For the resulting linear system to be solvable, at least $q + 1$ Mmesh vertices are needed. With only a few more vertices, the method is not robust, as it is observed for one case in Figure 3.11(d). Generally, this instability for least squares with too few data points is well known in the literature. It is explained by the Runge phenomenon for polynomial interpolation [15] and expected to be well-conditioned from $n < 2\sqrt{m}$ on, with n the polynomial order and m the number of data points [16]. For the case of $n^e_{\mathcal{M}} = 198$ and $n^e_{\mathcal{C}} = 20$, there are (in the mean) 9.9 Mmesh elements per Cmesh element. This correponds to $m = 10.9$ data points per least squares approximation. Therefore, one expects the approach to get ill-conditioned from $n \geq 2\sqrt{10.9} \approx 6.6$ on. This matches Figure 3.11(d). However, when evaluating the conditioning number of the linear system solved for the least squares approximation of the detoriating cases, it is only of order 10^3. With such a conditioning number one does not expect significant detoriations of the solutions, if the deviations in the system came only from numerical errors, not from any added noise. Thus, the instability of the least squares is not (only) explained by ill-conditioning. But within the frame of this thesis, no other reason was found.

In Figure 3.12, the $\widetilde{\mathrm{MpC}}$-convergence curves are shown. As the number of Cmesh elements $n^e_{\mathcal{C}}$ is fixed, refining the Mmesh increases the $\widetilde{\mathrm{MpC}}$. The geometric accuracy of the curving methods generally increases as well, up to the level of accuracy that can be obtained with the curving order q and $n^e_{\mathcal{C}}$. On this level, the curves stagnate. Further refinement of the Mmesh does not result in an improvement of the curved meshes. The optimal choice of Mmesh refinement therefore lies around the stagnation onset, in order to avoid both, an accuracy limitation as well as an over refinement that would lead to higher computational costs.

Fixed number of Mmesh elements

In a second study, the target curve is fixed with $n^e_{\mathcal{M}} = 998$. It is aimed to apply the curving methods for $n^e_{\mathcal{C}} \in \{6, 10, \ldots, 78\}$ and $q \in \{1, \ldots 15\}$, but limitations in the methods lead to

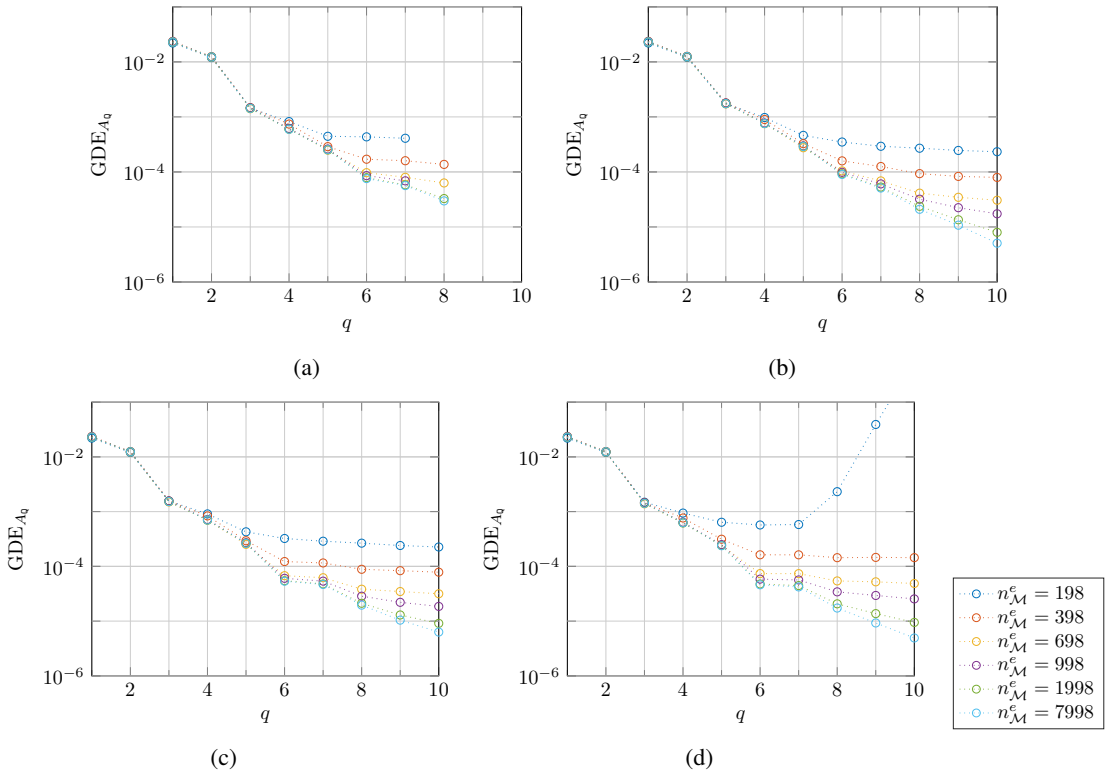


Figure 3.11: q-convergence for (a) the nodal method, (b) the reference based modal method, (c) the H^1 modal method, and (d) the least squares modal method with fixed $n^e_{\mathcal{C}} = 20$.

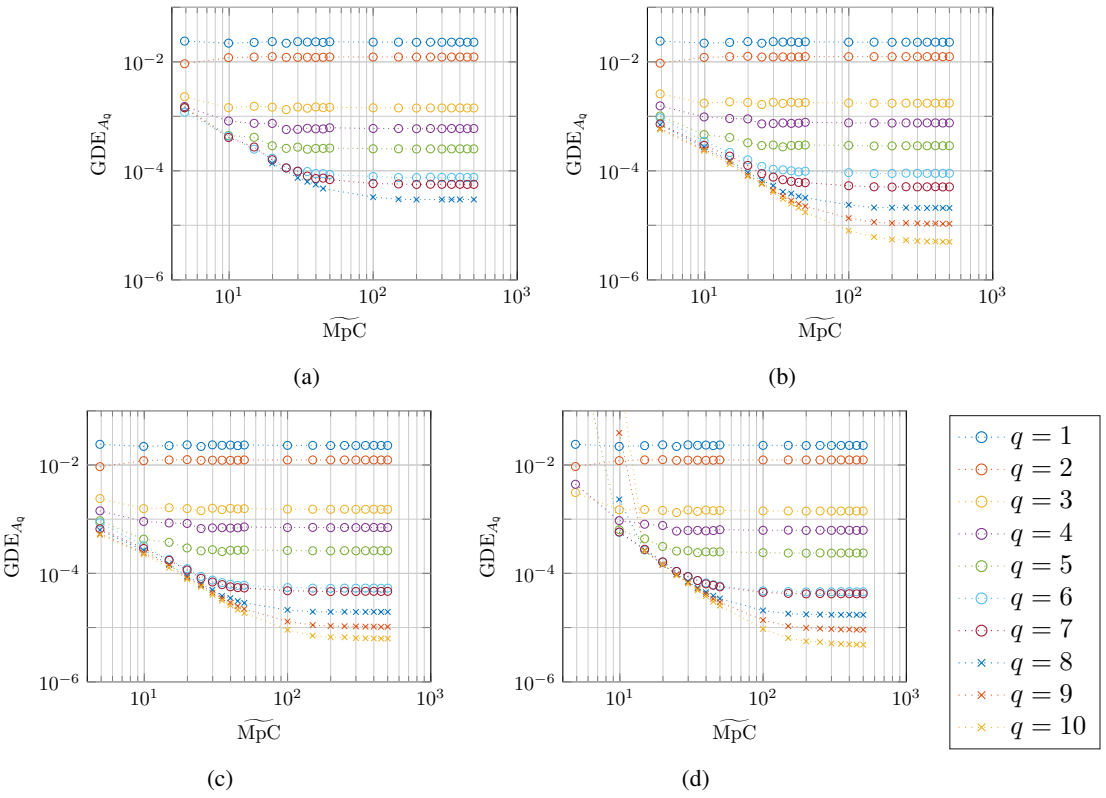


Figure 3.12: $\widetilde{\mathrm{MpC}}$-convergence for (a) the nodal method, (b) the reference based modal method, (c) the H^1 modal method, and (d) the least squares modal method with fixed $n^e_{\mathcal{C}} = 20$

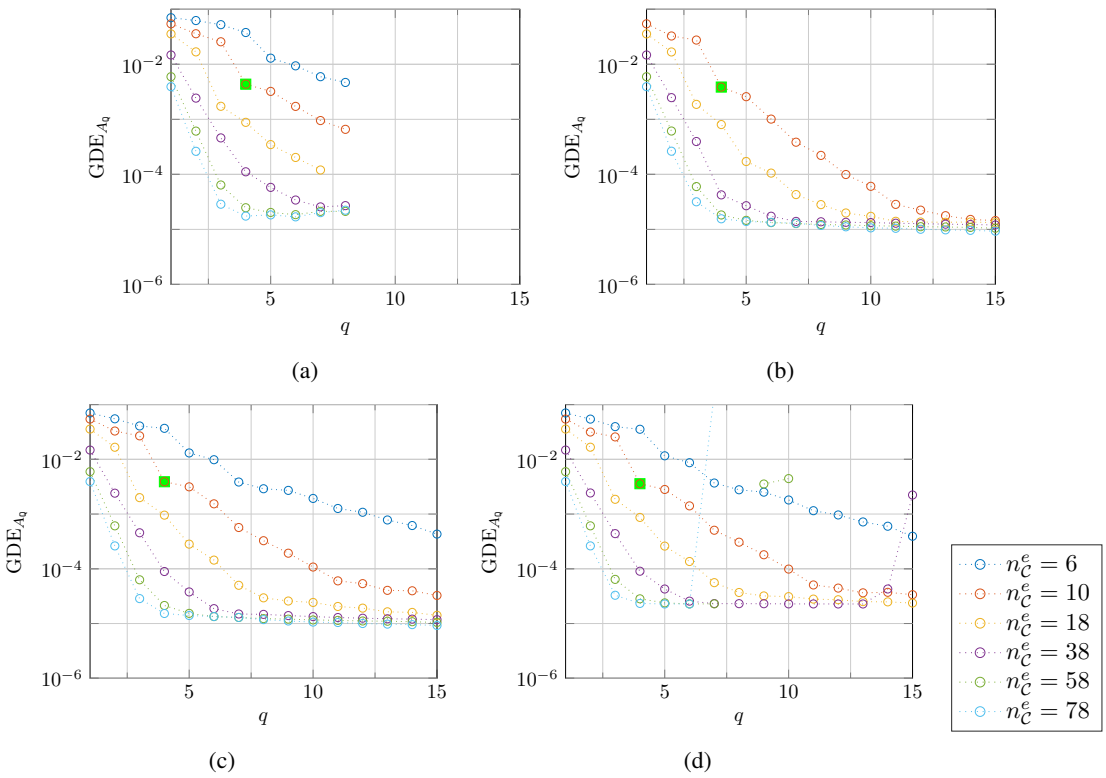


Figure 3.13: q-convergence for (a) the nodal method, (b) the reference based modal method, (c) the H^1 modal method, and (d) the least squares modal with fixed $n^e_{\mathcal{M}} = 998$.

restricted parameter sets:

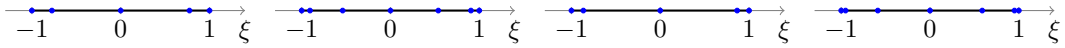
For the nodal curving, convergence issues occur for higher order approximations. The curving has to be limited to $q \leq 8$.

For the reference based modal curving, the limitations lie in the construction of the quadrangles that define the mapping from the reference to the physical space. For coarse $\mathcal{C}$, it could be hard or might not even be possible to find extrusion lengths that lead to quadrangles that are disjoint with their union containing all Mmesh vertices. As seen before, the least squares algorithm deteriorates for higher q if too few target points are available. Accordingly, high order approximations were not possible for relatively coarse Mmeshes. Only the H^1 modal approach can be applied to the full parameter range.

In Figure 3.13, the q-convergence curves for different $n^e_{\mathcal{C}}$ are shown. The green square highlights the example case from Figures 3.9(b)–3.10(d). All error curves decrease with increasing q and finally stagnate at the bound induced by the choice of $\mathcal{M}$. The size of $\mathcal{C}$ thereby controls the convergence rate before the onset of the stagnation. The finer the Cmesh is, the better is the approximation already for lower q values, and the sooner the accuracy bound is reached.

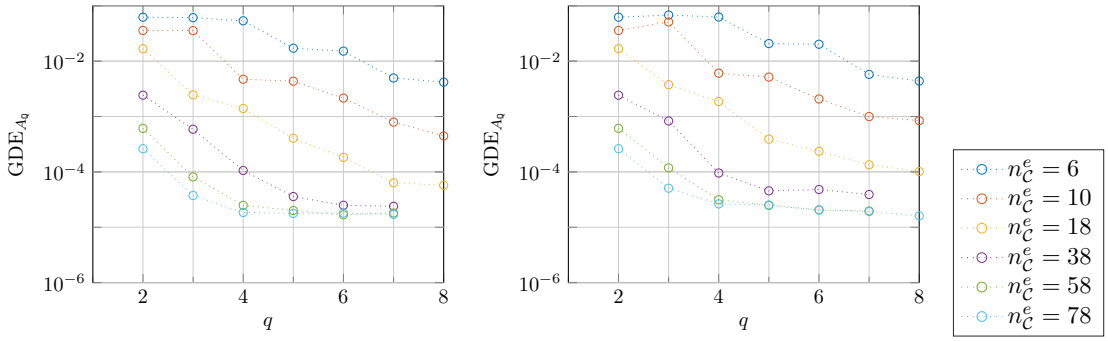
For the nodal curving algorithm, Legendre and Chebyshev node spacings in the reference space were tested to see if they provide better convergence results.

Legendre points are defined as the element vertices and $n_{\mathrm{int}} := (q+1) - 2$ interior nodes located at the roots of the n_{int}th Legendre polynomial. Similarly, interior Chebychev nodes are defined



(a) Legendre, q=4 (b) Legendre, q=6 (c) Chebyshev, q=4 (d) Chebyshev, q=6

Figure 3.14: Legendre and Chebyshev points based nodes.



(a) (b)

Figure 3.15: q-convergence for the nodal method with (a) Legendre or (b) Chebyshev nodes, with fixed $n_{\mathcal{M}}^e = 998$.

as the roots

$$t_i^{n_{\text{int}}} = \cos\left(\frac{(2i-1)\pi}{2n_{\text{int}}}\right)$$

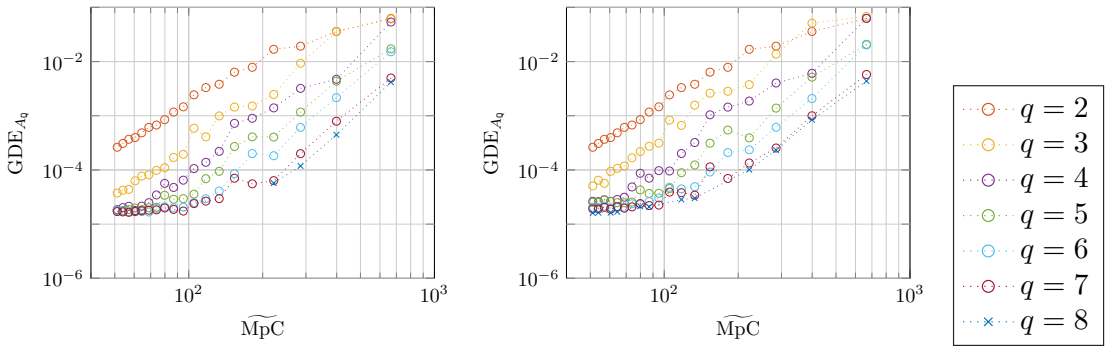
of the n_{int}th Chebychev polynomial of the first kind

$$T_n(t) = \cos(n \arccos(t)).$$

Both node types cluster towards the ends of the line element (see Figure 3.14) and therefore counteract oscillatory behaviour of the interpolation [67].

Corresponding to the change of the nodal location on the reference element, $\tilde{\boldsymbol{p}}_i$ in the relaxation step is not defined as the midpoint between $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_{i+1}$, but based on the ratio $\frac{\xi_i - \xi_{i-1}}{\xi_{i+1} - \xi_i}$.

Figures 3.15 and 3.16 show the q-convergence and ${}^{n_{\mathcal{M}}^e}/_{n_{\mathcal{C}}^e}$-convergence curves, respectively. There is no significant improvement of the convergence behaviour.



(a) (b)

Figure 3.16: $\widetilde{\text{MpC}}$-convergence for the nodal method with (a) Legendre or (b) Chebyshev nodes, with fixed $n_{\mathcal{M}}^e = 998$.

3.4 Conclusions of this Chapter

The four curving methods were applied to two test cases, the unit circle and a distorted ellipse. On the circle test case, the reference based method suffers from a convergence gap compared with the other three methods. For the more general geometry, this was not observed any more. Here, the requirements of the mapping from the reference to the physical space led to limitations in the choice of the mesh size. Besides this, it also did not provide significantly better results than the H^1 modal approach or the least squares.
For the nodal curving, the general problem lies in convergence issues within the curving process itself that make the whole approach very unstable. Tests with alternative node spacings did not resolve this.
All three methods that consider the full Mmesh as target encounter a stagnation of the error level due to the Mmesh accuracy. This especially includes the H^1 modal curving approach. A preprocessing step running a mesh reconstruction method for the Mmesh before the curving could improve this, similar to what is proposed in [13]. Otherwise, the H^1 modal curving seems to be quite robust and applicable to both tested geometries.
The least squares method bypasses this stagnation by using only exact points. This leads to much better results on relatively coarse meshes in the circle test case. On the more general geometry of the distorted ellipse, this advantage is less significant. Further, the least squares approach becomes unstable or even incalculable for too coarse $\mathcal{M}$. This effects especially those cases, where the H^1 modal approach stagnates due to the inaccuracy of the Mmesh and where the least squares approach could be the most beneficial. Note that here only a basic least squares is tested. Modifications of the method might lead to an improved stability.
Further note that the tests performed here are designed to compare the introduced curving methods. Besides the method comparison, only limited conclusions can be drawn. For the scenarios that motivated the method requirements, both the Mmesh and the Cmesh might be fixed - the Mmesh by the provided fine mesh and the Cmesh due to limitations on the mesh for simulation runs. In this case, only the curving order q is left as degree of freedom. The results of this chapter indicate that q refinement is only meaningful up to the point of stagnation onset which is geometry dependent and has therefore to be determined. Further, due to the stagnation, the target accuracy for the geometric description could not be chosen arbitrarily. If the desired accuracy lies below the stagnation level, an improvement of the Mmesh accuracy is needed, e.g. in form of the above mentioned mesh reconstruction methods.
From the two described test cases, no clear preference can be drawn among the considered curving methods. Both, the H^1 modal curving and the least squares approach look promising. In the following chapters, further tests will be run, considering the performance of the curved meshes in the context of acoustics simulations.

Chapter 4

Geometric Enhancement for Acoustic Simulations

In the previous two chapters, the description and generation of high-order meshes were discussed. Now the application of these high-order meshes to acoustic simulations is considered. Therefore, a short review of equations and methods in numerical acoustics is provided. The focus lies on the finite element method (FEM) and its high-order version p-FEM. Two geometric enhancements of the p-FEM will be described, of which the polynomial based qp-FEM will be chosen for the application to acoustic test cases in this and the subsequent chapters.

4.1 Review of Numerical Acoustics

In this section, some concepts on how to describe acoustic problems mathematically are summarised and how to solve them numerically.

4.1.1 Equations

Information herein is based on course notes [21]. The different equations to describe and solve acoustic problems are derived from the conservation of mass and the conservation of momentum by making varying assumptions, e.g. on the fluid and flow. As the acoustic perturbations are small compared to the ambient conditions, linearisation is often applied as well. Among the resulting equations are e.g. the Linearised Euler Equations for unsteady inhomogeneous flow in an inviscid fluid under adiabatic conditions [38]. Under the assumption of a homogeneous, inviscid fluid with no external flow and adiabatic conditions, and by linearisation, the linear acoustic wave equation

$$\nabla^2 \mathtt{p} - \frac{1}{c_0^2} \frac{\partial^2 \mathtt{p}}{\partial t^2} = -\rho_0 \frac{\partial \mathtt{q}}{\partial t}$$

is obtained with $\mathtt{p}$ the acoustic pressure field, ρ_0 the ambient mass density field, $\mathtt{q}$ the volume velocity source given as volume velocity per unit volume, and c_0 the speed of sound in the given fluid. As mentioned in the Introduction, this thesis examines acoustic simulations in the frequency domain. Therefore, the linear acoustic wave equation is considered for time harmonic excitations. These are present for tonal noise components or could be analytically obtained by Fourier tranformation. Assuming a harmonic time dependency (using $e^{+i\omega t}$ convention), the

linear acoustic wave equation simplifies to the Helmholtz equation

$$\nabla^2 \mathtt{p} + k^2 \mathtt{p} = -i\rho_0 \omega \mathtt{q} \tag{4.1}$$

with the acoustic wavenumber $k = \omega/c_0$ and angular frequency ω.

In order to completely define a problem and to obtain a unique solution, boundary conditions (BC) need to be introduced. The standard boundary conditions imposed on the boundary $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_R$ of the domain Ω and their physical meanings are the following:

- **Pressure**
  Dirichlet boundary conditions

  $$\mathtt{p}(x) = \mathtt{p}_{\Gamma_D}(x), \text{ for } x \in \Gamma_D$$

  describe the acoustic pressure field $\mathtt{p}_{\Gamma_D}$ imposed at Γ_D.

- **Normal velocity**
  By Neumann boundary conditions

  $$\nu_n(x) = \frac{i}{\rho_0 \omega} \frac{\partial \mathtt{p}(x)}{\partial n} = \nu_{n,\Gamma_N}(x), \text{ for } x \in \Gamma_N,$$

  the normal velocity ν_{n,Γ_N} along the boundary is prescribed with $\boldsymbol{n}$ the outwards facing normal of the domain Ω. Homogeneous Neumann BC correspond to a rigid wall, where the acoustic wave is completely reflected.

- **Impedance**
  Robin boundary conditions are also called mixed boundary conditions because they relate the field variable to its normal derivative. The relationship is described by a given normal impedance function Z_{Γ_R} that describes the damping properties of the material forming the boundary:

  $$\mathtt{p}(x) = Z_{\Gamma_R}(x)\nu_n(x) = \frac{iZ_{\Gamma_R}(x)}{\rho_0 \omega} \frac{\partial \mathtt{p}(x)}{\partial n}, \text{ for } x \in \Gamma_R.$$

- **Radiation condition**
  For exterior problems of dimension $\mathtt{d} \in \{2, 3\}$, additionally the Sommerfeld radiation condition

  $$\lim_{r \to \infty} r^{\frac{\mathtt{d}-1}{2}} \left(\frac{\partial \mathtt{p}}{\partial r} + ik\mathtt{p} \right) = 0$$

  with $r = |\boldsymbol{x}|$, $\boldsymbol{x} \in \Omega$ has to be fulfilled to ensure that acoustic waves propagate in free field without any reflections [84].

4.1.2 Numerical Methods

Analytical solutions are known for some acoustic problems with simple geometries, but are impossible to determine for complex geometries encountered in real-life applications. Instead,

numerical simulations are used to obtain an approximate solution of the problem. As stated in the Introduction, only mesh-based methods will be considered here.

4.1.2.1 Methods

In the *boundary element method (BEM)*, only the domain boundary needs to be discretised. It is a typical method to solve an exterior problem in an unbounded domain. The differential equation is reformulated as an integral formulation which is defined on the boundary of the domain. Accordingly, only a discretisation of the boundary is needed which simplifies the meshing process and leads to a comparatively small number of degrees of freedom (DoFs). The resulting linear system is complex and dense, which precludes the application of efficient solving algorithms for sparse systems. In a post-processing step, the solution is expanded to the interior of the domain. Due to the need of a Green's function for the construction of the integral formulation, the applicability of the BEM for inhomogeneous or complex domains is restricted ([88], [21]).
In these cases, or for problems on bounded domains, domain based methods are more suitable [21]. Among these are *finite difference methods (FDM)*. Derivatives of the field variable are approximated by difference schemes that are obtained from Taylor expansions over stencils around the evaluation node. The evaluation relies on structured grids and is therefore unsuitable for complex geometries [26].
Methods on unstructured grids are typically based on the weak formulation of the problem. Based on a weighted residual method, a weak formulation of the problem is obtained by applying a scalar product with a test function φ to both sides of the differential equation. It can be proven that solutions of the strong formulation are also solutions of the weak formulation, if the weak formulation holds for all $\varphi \in V$.
For the discretisation, the solution space V is first restricted to a finite-dimensional subspace V_h. Then the solution is expressed as a linear combination of a suitable basis of V_h. The basis functions are also called *shape functions*. By substituting the φ with actual test functions in V_h, the continuous problem is transformed in a discrete system of equations. Usually, the shape functions are used as test functions.
If V_h is chosen as a subspace of continuous functions, a *(continuous) Galerkin method* is obtained. For the *discontinuous Galerkin method (DGM)*, the functions in V_h do not need to be continuous. This method allows for functions which are only piecewise continuous, with discontinuities along element boundaries.

4.1.2.2 Standard FEM

Finite element methods (FEMs) are continuous Galerkin methods. The common choice of the solution space V for FEMs is the Sobolev space $H^1(\Omega)$, defined as

$$H^1(\Omega) := \left\{ v \in L^2(\Omega) \middle| \frac{\partial v}{\partial x_i} \in L^2(\Omega), 1 \leq i \leq \mathtt{d} \right\},$$

with Ω the domain on which the method is formulated, $\mathtt{d}$ the dimension of the problem, and $L^2(\Omega)$, the space of functions that are square integrable over Ω [82]. For simplicity, the argument

Ω will be omitted from here on. Note the following definitions of the L^2-norm $||\cdot||_{L^2}$, the H^1-seminorm $|\cdot|_{H1}$, and the H^1-norm $||\cdot||_{H^1}$ that will be used in the remainder of this thesis:

$$\begin{aligned} ||v||_{L^2} &:= \left(\int_\Omega v\bar{v}\,\mathrm{d}\Omega\right)^{1/2}, \\ |v|_{H^1} &:= ||\nabla v||_{L^2}, \\ ||v||_{H^1} &:= ||v||_{L^2} + ||\nabla v||_{L^2}, \end{aligned}$$

where $\bar{v}$ denotes the complex conjugate of v. The corresponding scalar products of L^2 and H^1 are

$$\begin{aligned} \langle u, v\rangle_{L^2} &:= \int_\Omega u\bar{v}\,\mathrm{d}\Omega, \\ \langle u, v\rangle_{H^1} &:= \langle u, v\rangle_{L^2} + \langle \nabla u, \nabla v\rangle_{L^2}\,. \end{aligned}$$

To obtain the weak formulation, the scalar product $\langle\cdot, \varphi\rangle_{L^2}$ is applied to the problem. For the example of the Helmholtz equation (4.1), this leads to the weak formulation

$$\begin{aligned} \int_\Omega \bar{\varphi}\nabla^2\mathtt{p} + k^2\bar{\varphi}\mathtt{p}\,\mathrm{d}\Omega &= \int_\Omega -i\rho_0\omega\bar{\varphi}\mathtt{q}\,\mathrm{d}\Omega \\ \Leftrightarrow \int_\Omega \nabla\bar{\varphi}\cdot\nabla\mathtt{p}\,\mathrm{d}\Omega - \int_\Gamma \bar{\varphi}\frac{\partial\mathtt{p}}{\partial n}\,\mathrm{d}\Gamma - k^2\int_\Omega \bar{\varphi}\mathtt{p}\,\mathrm{d}\Omega &= i\rho_0\omega\int_\Omega \bar{\varphi}\mathtt{q}\,\mathrm{d}\Omega. \end{aligned} \tag{4.2}$$

Terms depending on both the field variable $\mathtt{p}$ and the test function φ are on the left-hand side, whereas terms on the right-hand side only depend on φ.

If Neumann or Robin BC are specified, they are directly included in the second term in (4.2) over the domain boundary Γ. For homogeneous Neumann BC, the term vanishes, otherwise it becomes dependent only on φ and can therefore be written on the right-hand side of the formulation. With Robin BC, the term still depends on $\mathtt{p}$ and therefore remains on the left-hand side. For simulations on unbounded domains, an artificial boundary has to be introduced. Special techniques, such as the *perfectly matched layer (PML)* [11] or Robin boundary conditions called *characteristic boundary conditions* [87], are available to limit spurious reflections and fulfil the Sommerfeld radiation condition [88].

The *standard FEM* is obtained by choosing the space of continuous, piecewise linear functions for V_h^1 on Ω

$$V_h^1 = \left\{ v_h \in V \,\middle|\, v_h \text{ continuous},\ v_h|_{\Omega_i} \in \mathbb{P}^1,\ \dot{\bigcup_i}\Omega_i = \Omega \right\},$$

with $\mathbb{P}^p$ the space of polynomials of order p and $\dot{\bigcup}_i\Omega_i$ the union of disjoint subsets Ω_i of Ω. The basis and test functions are formed by piecewise linear shape functions N_i with locally bounded support. Each of these shape functions N_i is associated with one vertex $\boldsymbol{x}_i$ of the mesh that triangulates Ω. They are defined such that

$$N_i(\boldsymbol{x}_j) = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{for } i \neq j. \end{cases}$$

Since N_i is piecewise linear, it follows that its support is restricted to the elements containing $\boldsymbol{x}_i$. These shape functions are defined over the whole domain and are therefore called *global shape functions*.

Using $\{N_i\}_{i=1}^{n_N}$ as basis of V_h^1, the piecewise linear approximation $\mathtt{p}_h \approx \mathtt{p}$ with $\mathtt{p}_h \in V_h^1$ can be written as the linear combination

$$\mathtt{p}_h = \sum_{i=1}^{n_N} \mathtt{p}(\boldsymbol{x}_i) N_i = \sum_{i=1}^{n_N} \mathtt{p}_i N_i, \tag{4.3}$$

where n_N denotes the number of global shape functions, and $\mathtt{p}_i := \mathtt{p}(\boldsymbol{x}_i)$ coefficients that need to be determined. Remember that Dirichlet BC were not built into the weak formulation. Instead, they are included in the method formulation at this point. For each node $\boldsymbol{x}_i$ that falls on the domain of Dirichlet BC, the coefficient $\mathtt{p}_i$ can be directly determined from the BC. Therefore only $n_N - n_D$ of unknowns are left to be determined, with n_D the number of nodes covered by the Dirichlet BC.

In (4.2), $\mathtt{p}$ is substituted by (4.3) and each global shape function that is associated with a node that lies not in a Dirichlet BC domain is applied individually as test functions φ. This results in $n_N - n_D$ equations of the form

$$\begin{aligned}
&\int_\Omega \nabla N_j \sum_{i=1}^{n_N} \mathtt{p}_i \nabla N_i \,\mathrm{d}\Omega - k^2 \int_\Omega N_j \sum_{i=1}^{n_N} \mathtt{p}_i M_i^e \,\mathrm{d}\Omega + \frac{i\rho_0\omega}{Z_{\Gamma_R}} \int_{\Gamma_R} N_j \sum_{i=1}^{n_N} \mathtt{p}_i M_i^e \,\mathrm{d}\Gamma_R \\
&\quad = -i\rho_0\omega \int_{\Gamma_N} N_j \nu_{n,\Gamma_N} \,\mathrm{d}\Gamma_N + i\rho_0\omega \int_\Omega N_j \mathtt{q} \,\mathrm{d}\Omega \\
\Leftrightarrow &\sum_{i=1}^{n_N} \mathtt{p}_i \int_\Omega \nabla N_j \nabla N_i \,\mathrm{d}\Omega - k^2 \sum_{i=1}^{n_N} \mathtt{p}_i \int_\Omega N_j N_i \,\mathrm{d}\Omega + \frac{i\rho_0\omega}{Z_{\Gamma_R}} \sum_{i=1}^{n_N} \mathtt{p}_i \int_{\Gamma_R} N_j N_i \,\mathrm{d}\Gamma_R \\
&\quad = -i\rho_0\omega \int_{\Gamma_N} N_j \nu_{n,\Gamma_N} \,\mathrm{d}\Gamma_N + i\rho_0\omega \int_\Omega N_j \mathtt{q} \,\mathrm{d}\Omega
\end{aligned} \tag{4.4}$$

that determine the $n_N - n_D$ unknowns $\mathtt{p}_i$. Note that the term over Γ was substituted by the two terms over Γ_N and Γ_R for Neumann and Robin BC, respectively. Due to the local support of N_i, the method results in a sparse system of linear equations, which allows for fast and memory efficient computation.

To implement the FEM, it is favourable to assemble the system (4.4) elementwise. Thereby, the integrations over Ω are split up into integrations over single elements, and the integration over Γ into integrals over single boundary elements. The use of linear parametric mappings as described in Section 2.1.1.1 enables integrations over the straight sided physical elements T to be transformed and computed on the reference element $\hat{T}$, using the equation

$$\int_T f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int_{\hat{T}} f(\boldsymbol{F}(\boldsymbol{\xi})) |J_{\boldsymbol{F}}(\boldsymbol{\xi})| \,\mathrm{d}\boldsymbol{\xi}, \tag{4.5}$$

with the mapping $\boldsymbol{F} : \hat{T} \to T$ and $J_{\boldsymbol{F}}$ the Jacobian of $\boldsymbol{F}$, which corresponds to the determinant of the Jacobian matrix. If the integrand contains a gradient, this gradient also needs to be

transformed to the reference coordinate by the substitution

$$\nabla_{\boldsymbol{x}} = J_{\boldsymbol{F}}^{-1} \nabla_{\boldsymbol{\xi}}. \tag{4.6}$$

This transformation of the integration onto $\hat{T}$ further simplifies the implementation, since quadrature rules must only be given for $\hat{T}$.
On the reference element $\hat{T}$, *local shape functions* N_i^e are considered. Each N_i^e is associated with one of the reference element vertices $\boldsymbol{\xi}_i$ and is defined such that

$$N_i^e(\boldsymbol{\xi}_j) = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{for } i \neq j \end{cases}.$$

These local shape functions correspond to the linear Lagrange shape functions defined in Section 2.1.1.2.

In [43], an error estimate in the H^1-seminorm for the Helmholtz equation solved with the standard FEM with linear shape functions is given as

$$E_1 \leq c_1 kh + c_2 k^3 h^2, \tag{4.7}$$

with constants c_1 and c_2 independent of the mesh size h and the wavenumber k. The first term expresses the error arising from the linear interpolation of the solution. It remains constant, if the non-dimensional wavenumber kh is kept fixed. Therefore, the rule of thumb of 10 elements per wavelength is typically applied in the literature ([20], [88]).
By the second term in (4.7), the pollution effect [7] is represented. In FEM simulation results, the wavenumber $\tilde{k}$ differs from the wavenumber k of the analytical solution. The relative difference between k and $\tilde{k}$ is called *dispersion error*. It leads to a phase shift between the analytical and the numerical solution which adds up for each wavelength. It is therefore especially prominent on long simulation domains and becomes more and more severe as the frequency increases.
In order to keep the pollution effect small for standard FEM simulations, a constant kh is not sufficient. As k increases, the mesh must be further refined. This is computationally prohibitive and limits the use of the standard FEM approach to the low frequency range. Several techniques exist to avoid this issue and to increase the computable frequency range. Among these are the stabilised Galerkin methods and the wave-based discretisation methods [88]. Increasing the field interpolation order can also be used to reduce the pollution effect [9]. The resulting high-order FEM is called *p-FEM* and will be considered from here on.

4.1.2.3 High-order p-FEM

For high-order p-FEM, the space V_h^p is not chosen as piecewise linear functions, but instead as the space of piecewise polynomial functions of order p:

$$V_h^p = \left\{ v_h \in V \;\middle|\; v_h \text{ continuous},\ v_h\big|_{\Omega_i} \in \mathbb{P}^p,\ \dot{\bigcup_i} \Omega_i = \Omega \right\}.$$

The error estimate in Equation (4.7) generalises to

$$E_1 \leq c_1 \left(\frac{kh}{2p}\right)^p + c_2 k \left(\frac{kh}{2p}\right)^{2p} \tag{4.8}$$

[7, 44, 42]. Commonly, high-order Lagrange basis functions as those used for the geometric description in Section 2.1.1.3 are used as local shape functions in $\hat{T}$. In this thesis, Lobatto shape functions as defined in Section 2.1.1.4 are chosen as basis and test functions instead.

Since bubble shape functions are only non-zero in the interior of elements, static condensation can be applied. Thereby, the contribution of internal degrees of freedom (DoFs) are removed from the linear system. This reduces the size of the system to be solved and improves its conditioning. The solution for the internal DoFs is locally calculated in a post-processing step from the solution of the external DoFs of the corresponding element [82]. Further, their hierarchical behaviour allows for efficient p-adaptivity algorithms [9].

4.2 Review on Geometry Enhanced Implementations

Both, standard FEM and high-order p-FEM are generally based on linear, straight sided meshes. This section summarises the incorporation of enhanced geometric representations for the p-FEM.

In the literature, there are two main ways how enhanced geometric representations are included into p-FEM implementations. The standard way is the high-order polynomial description. The second described way is the rather recently developed NURBS based geometric description of the NURBS-enhanced FEM (NEFEM) method.

4.2.1 High-order Polynomial Geometric Description

As mentioned in Section 4.1.2.2, parametric mappings from T to $\hat{T}$ are used to transform the integration over T onto $\hat{T}$. As described in that section, they form the geometric description of the approximation of the domain by piecewise linear elements. For higher-order geometric descriptions of the domain, the order q of the parametric mapping has to be increased. Therefore, the linear reference elements and linear geometric shape functions are replaced by the corresponding higher-order reference elements and shape functions as defined in Section 2.1.1.1. If Lagrange shape functions are used, higher-order nodes need to be included in the physical mesh. As seen in Section 2.1.1.4, this is not necessary if Lobatto shape functions are applied.

Note that the choices of the field polynomial basis and the geometrical polynomial basis are independent of each other. The same applies to the choices of the field interpolation order p and the geometric interpolation order q. Therefore, this kind of methods is denoted as *qp-FEM* from here on. Further note that due to the non-linearity of the mapping $\boldsymbol{F}$ for higher-order geometric interpolation, the Jacobian is not constant for non-straight sided elements. For integrals with gradients this leads to non-polynomial integrands, that cannot be integrated exactly using Gaussian quadrature.

4.2.2 NURBS-based Geometric Description

An approach to embed a CAD-based description of the domain geometry within the known FEM setup is given by the *NURBS-enhanced FEM (NEFEM)* ([79], [81]). Contrarily to isogeometric analysis (IGA) where the whole domain and the field are described using NURBS, the NEFEM approach uses the CAD information only to describe the boundary curves or surfaces. Interior edges or faces of the mesh remain linear, and polynomial interpolation based on Lagrange polynomials is used for the field representation. This way, the advantage of IGA, namely exact representation of CAD surfaces, is combined with the computational advantages of FEM. Integration over elements is done sectionwise, where the line or triangle element is subdivided at breakpoints or along knot lines respectively, to also include the possibility of reduced continuity order within elements ([78], [80]).

Similarly to the polynomial case, the integration is performed on reference elements. For interior elements not affected by the NURBS boundary representation, the standard procedure shown in Figure 2.1 is applied. Integration over elements with an edge along the NURBS boundary is evaluated on a rectangular reference element. To take into account the parametrisation changes at breakpoints, the reference element is subdivided accordingly. The quadrature is applied to each part of the reference element separately.

Since this thesis is based on the assumption that no CAD description is available and since the curving algorithms described in Chapter 3 provide polynomially curved meshes, this approach is not applicable. Instead, a polynomial based geometric description is used.

4.3 Embedding Curved Elements into pFEM

For the 2-dimensional curving approaches described in Sections 3.1 and 3.2, only boundary edges are considered. As a subsequent step, the curved edges must be embedded into the surface mesh. Therefore, a mapping from a reference element to the element with the curved edge in the physical space is defined. From here on, triangular meshes are considered.

4.3.1 Nodal Elements

In order to embed curved edges into nodal triangular elements, the geometric order of the latter has to be enhanced to the order of the curved edge. Therefore, high-order nodes are added along the domain-interior edges. For complete elements (as described in 2.1.1.3) of order $q > 2$, further high-order nodes have to be placed in the element interior.

In the *Gmsh* implementation, edge nodes are placed equidistantly. Together with the original vertex nodes, they form serendipity (also called incomplete) elements ([86]). Element internal nodes are then placed on the high-order interpolation with the shape functions of the serendipity element (see [35]).

3-dimensional elements are treated analogously by first mapping the faces and then placing the interior volume nodes based on the shape functions of the incomplete 3-dimensional element.

4.3.2 Modal Elements, Blending Approach

In order to embed the modally curved edges into a 2-dimensional mesh, a blending approach is proposed. It is derived from the construction of the Lobatto shape functions (see Section 2.1.1.4). Each triangular element is represented by a mapping from the reference triangle $\hat{T}^1_{\text{tria}}$ to the physical space. Standard linear mappings $\boldsymbol{F}(\xi,\eta) = \sum_{i=1}^{3} \boldsymbol{x}_i \varphi_T^{v_i}(\xi,\eta)$ that are defined by the vertices $\boldsymbol{x}_i$ are used for the straight sided elements. For elements with a curved edge, the mapping has the form

$$\boldsymbol{F}(\xi,\eta) = \sum_{i=1}^{3} \boldsymbol{x}_i \varphi_T^{v_i}(\xi,\eta) + \sum_{i=1}^{3} \boldsymbol{\kappa}^{e_i}(\xi,\eta),$$

where $\boldsymbol{\kappa}^{e_i}$ corresponds to the edge e_i. It is defined as

$$\boldsymbol{\kappa}^{e_i}(\xi,\eta) = \begin{cases} \boldsymbol{0} & \text{on } e_j,\ i \neq j \\ \boldsymbol{d}^{e_i}\left(\tilde{\xi}^{e_i}(\xi,\eta)\right) & \text{on } e_i \end{cases}, \tag{4.9}$$

with the auxiliary variable $\tilde{\xi}^{e_i}(\xi,\eta) = \lambda_{i+1}(\xi,\eta) - \lambda_i(\xi,\eta) \in [-1,1]$, λ_i the barycentric coordinates of the triangle (see (2.3)), $\lambda_4 = \lambda_1$, and

$$\boldsymbol{d}^{e_i}\left(\tilde{\xi}^{e_i}\right) = \boldsymbol{F}_c^{e_i}\left(\tilde{\xi}^{e_i}\right) - \boldsymbol{F}_s^{e_i}\left(\tilde{\xi}^{e_i}\right),$$

where $\boldsymbol{F}_s^{e_i}$ describes the straight sided edge and $\boldsymbol{F}_c^{e_i}$ the actual, possibly curved edge.
For the H^1 modal mesh curving and the construction of the Lobatto shape functions it follows that for the x-component

$$\kappa_x^{e_i}(\xi,\eta) = \sum_{j=2}^{q} \varphi_{T,j}^{e_i}(\xi,\eta)\, \alpha_{x,j}^{e_i},$$

with $\alpha_{x,j}^{e_i}$, the α-coefficient of the triangle edge shape function $\varphi_{T,j}^{e_i}$ (see Equation (2.6)) for the x-component of edge e_i. The construction for the y-component is done analogously.
The blending approach provides more flexibility than the nodal based embedding of the curved edges. In general, it could also be applied to edges that are curved in another way than the H^1 modal curving approach introduced in Section 3.2.2.3. An assumption thereby is that $\boldsymbol{\kappa}^{e_i}$ is continuous. In Appendix A, the blending function $\boldsymbol{\kappa}$ for edges curved with the reference based modal approach is derived.
As mentioned in Sections 4.1.2.2 and 4.2, the reference mappings $\boldsymbol{F}$ of the physical mesh elements for the elementwise integration on the reference element form the geometric representation of the simulation domain. They enter the simulation equations in form of their Jacobians in Equations (4.5) and (4.6).

4.4 Validation of the Embedding

In this section, the embedding process described above is validated with a scattering test case. Results on meshes that were curved with the H^1 modal curving or the least squares approach (see

Section 3.2.2) and embedded as described above are compared with the results obtained on a reference nodal high-order mesh that was created using *Gmsh*. This test case is part of the paper [99].

4.4.1 Test Case: Scattering of a Plane Wave by a Cylinder (2D)

To test the embedding process, the scattering of a plane wave by a cylinder is considered. It is a simple 2D test case with a known analytical solution p_{ex} and therefore suitable for first tests. The scattering test case is given as

$$\begin{aligned} \nabla^2 \mathrm{p} + k^2 \mathrm{p} &= 0, \\ \frac{\partial \mathrm{p}}{\partial n} &= 0 \quad \text{on } \Gamma_s, \\ \frac{\partial \mathrm{p}}{\partial n} &= \frac{\partial \mathrm{p}_{\mathrm{ex}}}{\partial n} + \mathrm{i}\omega \mathrm{p}_{ex} - \mathrm{i}\omega \mathrm{p} \quad \text{on } \Gamma_o. \end{aligned} \tag{4.10}$$

Characteristic BC are applied on the outer boundary Γ_o. The domain Ω with Γ_o and scattering boundary Γ_s is shown in Figure 4.1 and some exemplary meshes in Figure 4.2.

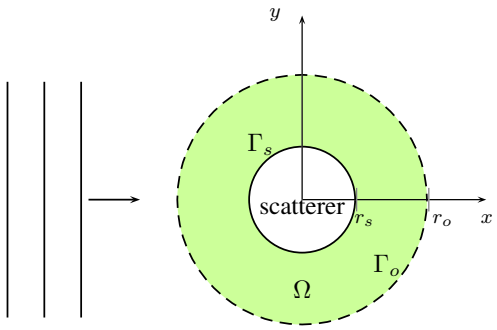


Figure 4.1: Sketch of the scattering test case.

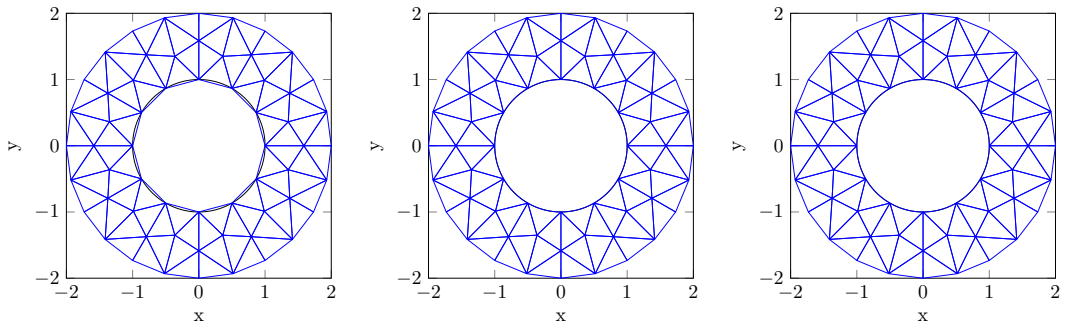
Characteristic boundary conditions

Equation (4.10) shows the formula for the characteristic boundary conditions on the outer boundary Γ_o. For the calculation of $\nabla_{x,y}\mathrm{p}_{\mathrm{ex}}$, the exact solution in $e^{-i\omega t}$ convention $\mathrm{p}_{\mathrm{ex}}^{-}$ is provided in [8]. The complex conjugate is taken of the resulting formula in order to match the $e^{+i\omega t}$ convention.

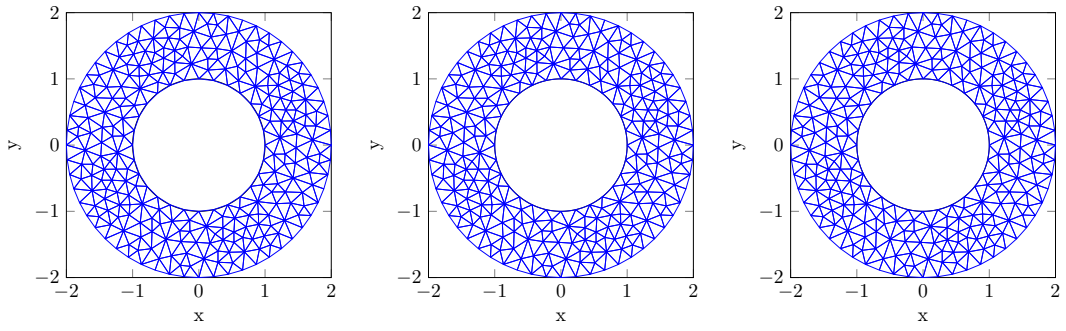
Weak formulation

The weak formulation with the weighting function φ is

$$\begin{aligned} \int_\Omega \nabla\bar{\varphi}\nabla\mathrm{p}\,\mathrm{d}\Omega - k^2 \int_\Omega \bar{\varphi}\mathrm{p}\,\mathrm{d}\Omega &= \int_\Gamma \bar{\varphi}\frac{\partial \mathrm{p}}{\partial n}\,\mathrm{d}\Gamma \\ &= \int_{\Gamma_o} \bar{\varphi}\left(\frac{\partial \mathrm{p}_{ex}}{\partial n} + \mathrm{i}\omega \mathrm{p}_{ex} - i\omega \mathrm{p}\right)\,\mathrm{d}\Gamma_o, \\ \Rightarrow \int_\Omega \nabla\bar{\varphi}\nabla\mathrm{p}\,\mathrm{d}\Omega - k^2 \int_\Omega \bar{\varphi}\mathrm{p}\,\mathrm{d}\Omega + \mathrm{i}\omega \int_{\Gamma_o} \bar{\varphi}\mathrm{p}\,\mathrm{d}\Gamma_o &= \int_{\Gamma_o} \bar{\varphi}\left(\frac{\partial \mathrm{p}_{ex}}{\partial n} + \mathrm{i}\omega \mathrm{p}_{ex}\right)\,\mathrm{d}\Gamma_o, \end{aligned}$$



(a) $n_{\mathcal{C}}^e = 12, q = 1$ (b) $n_{\mathcal{C}}^e = 12, q = 4$ (c) $n_{\mathcal{C}}^e = 12, q = 4$



(d) $n_{\mathcal{C}}^e = 28, q = 1$ (e) $n_{\mathcal{C}}^e = 28\ q = 4$ (f) $n_{\mathcal{C}}^e = 28, q = 4$

Figure 4.2: Linear meshes (a),(d) and 4^{th} order meshes curved with (b), (e) the H^1 modal approach and (c), (f) the least squares curving, for different $\mathcal{C}$-values of the circle scattering test case with $\mathrm{MpC} = 5000$. Note that for the curved meshes, only the interior circular boundary, the scatterer, is curved.

where $\Gamma = \Gamma_s \cup \Gamma_o$ is the boundary of Ω and R the radius of Γ_o.
For the numerical integration, a Gauss-Legendre quadrature is applied. In numerical tests, the order of the quadrature rule was found to be sufficient with $2p + 2(q-1)$ for element integrals and $p + (q-1)$ for boundary integrals. As safeguard, the quadrature orders are set to $2p + 2(q-1) + 4$ and $p + (q-1) + 4$, respectively. Note that based on [5] and [22], the integration order for curved geometries should be increased by $p - 1$ to preserve the exponential convergence. For $q \geq p - 1$, our chosen quadrature order lies above the theoretically derived one. In the region of $q < p - 1$, the geometric order is not high enough to ensure exponential convergence anyway (see also [5], [22]). There, the error induced by the geometry is the dominating error and the integration error can be neglected.

4.4.2 Evaluation of the Field Error

The standard error evaluation procedure for parametric FEM is the elementwise integration of the difference between the numerical and the analytical solution. Thereby, the numerical solution is interpolated using Gaussian quadrature with quadrature points on $\hat{T}$. The corresponding analytical solution is evaluated at the quadrature points mapped by geometric interpolation of the nodes of T. From [97] it is clear that this mapping does not map on the actual physical space Ω, but on its approximation $\hat{\Omega}$. Since $\hat{\Omega}$ is generally not a subset of Ω, the quadrature points could be mapped outside of Ω. For the scattering test case at the cylinder, this would lead to evaluations of the analytical solution inside of the scatterer where the analytical solution is not defined (see

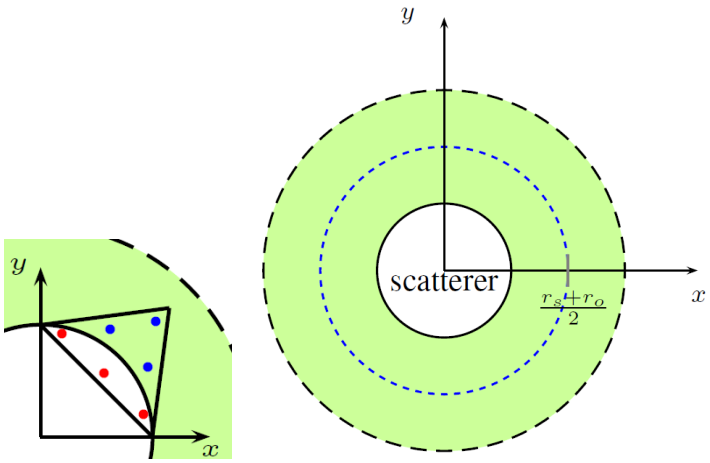
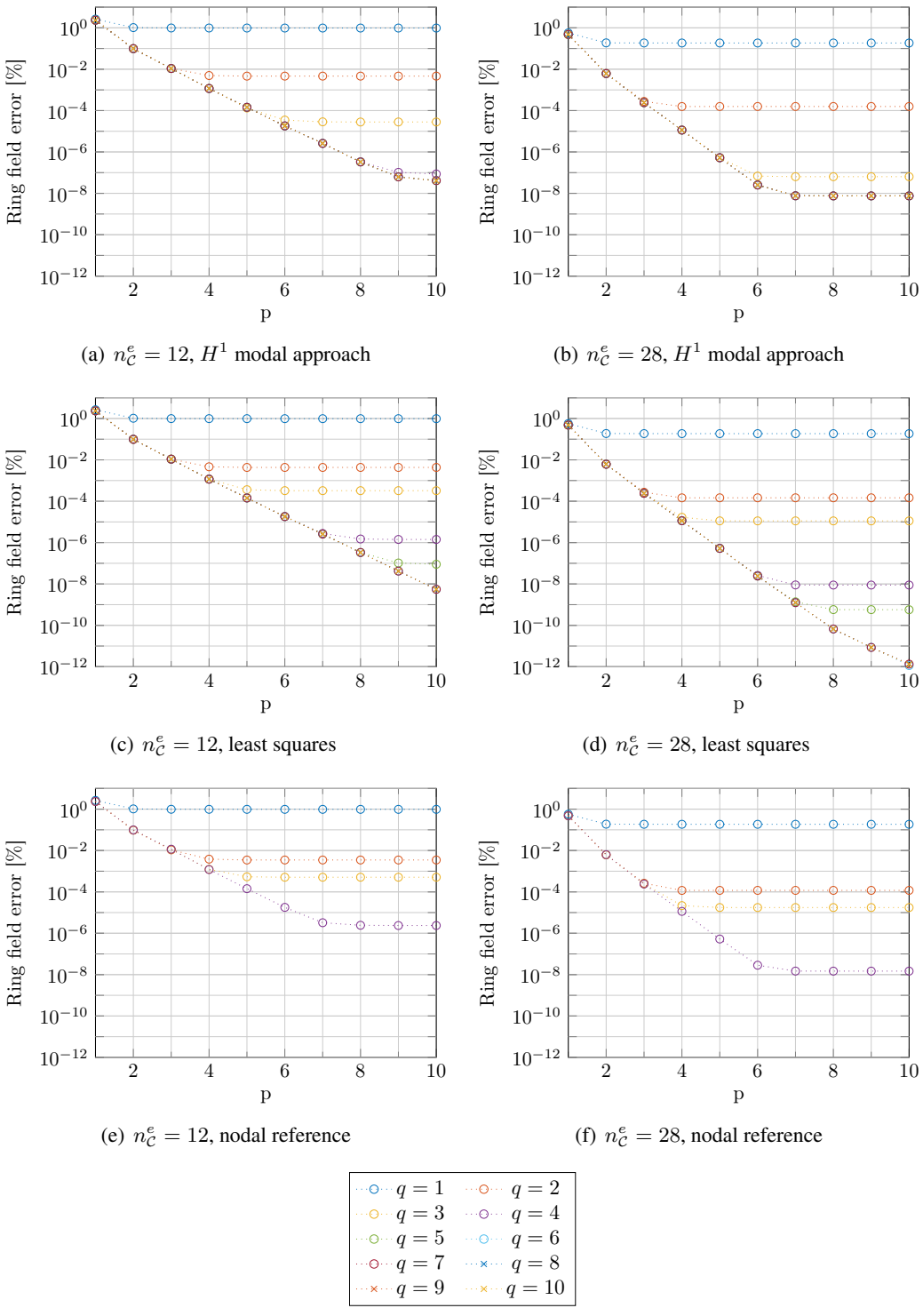


Figure 4.3: Schematic figures showing (a) the problem in the standard field error evaluation for the scattering test case and (b) the alternative approach. The red dots in (a) are the quadrature points that fell out of Ω.

Figure 4.3(a)). To avoid this problem, an alternative approach is considered, where the field error is evaluated as the relative L^2-error along a ring in the middle of the radial direction of Ω (see Figure 4.3(b)). This error measure is denoted as $E_{2,R}$. With this approach it is ensured that none of the points along the line integral lie outside of the physical domain. Since the mesh elements do not follow this interior line, the integrals are evaluated numerically with a trapezoidal rule. An adaptive refinement of the integration points is applied until a relative accuracy of 10^{-8} or an absolute accuracy of 10^{-15} is reached.
The choice of this field error measure will be confirmed in the following Chapter, where also two other approaches are considered. Further note that the dimension reduction (when changing from an area measure to a measure evaluated along a curve only) does not affect the expected exponential convergence (see [5], [22], or Equ. 5.1). The formula for the convergence is independent of the dimension.

4.4.3 Results

The simulation results are presented as p-convergence curves in Figures 4.4 - 4.6. Results on nodally curved meshes that were obtained with *Gmsh* are provided as reference. The latter are provided only up to $q = 4$. This limitation is not based on *Gmsh*, but on the simulation code. As the nodal shape functions were implemented for each order separately, this was done only up to order 4. A better way would have been to apply the computation of Lagrange shape functions described in [39]. It is based on Jacobi Polynomials and applicable for arbitrary polynomial orders and nodal distributions. In general, all curves show a similar behaviour: With increasing interpolation order p the simulation error decreases. For lower p, the curves for different q orders collapse. One by one, the error curves stagnate, with curves corresponding to lower q orders stagnating earlier than those of higher q.
The most interesting part is the stagnation value. As further field refinement does not lead to any improvement of the simulation result, the stagnation level indicates the error introduced by the discretisation of the geometry. Here, this error is dominated by the geometric inaccuracy of the



(a) $n_{\mathcal{C}}^{e} = 12$, H^1 modal approach

(b) $n_{\mathcal{C}}^{e} = 28$, H^1 modal approach

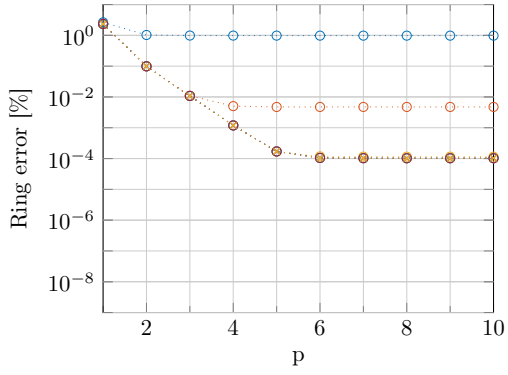
(c) $n_{\mathcal{C}}^{e} = 12$, least squares

(d) $n_{\mathcal{C}}^{e} = 28$, least squares

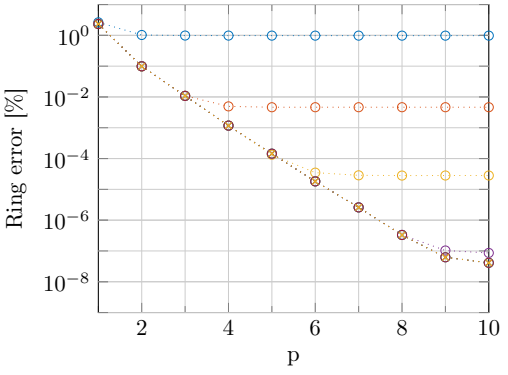
(e) $n_{\mathcal{C}}^{e} = 12$, nodal reference

(f) $n_{\mathcal{C}}^{e} = 28$, nodal reference

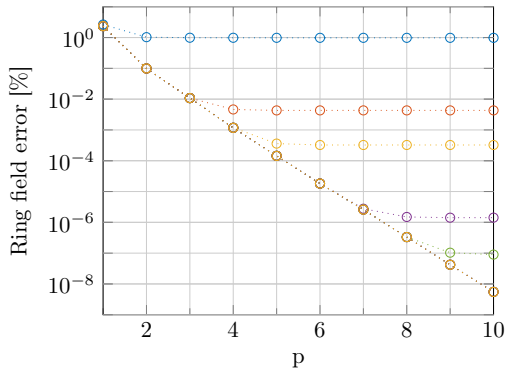
Figure 4.4: Comparing p-convergence curves for different $n_{\mathcal{C}}^{e}$-values of the circle scattering test case with $\mathrm{MpC} = 5000$ and $\omega = 1$.



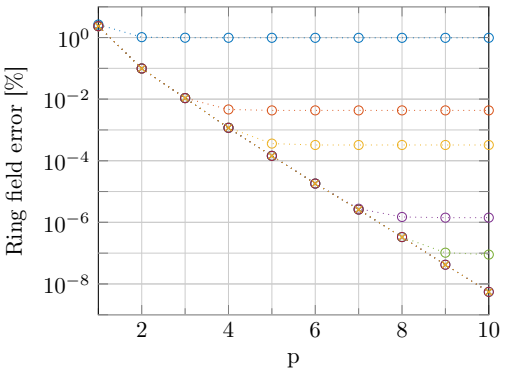
(a) $MpC = 100$, H^1 modal approach



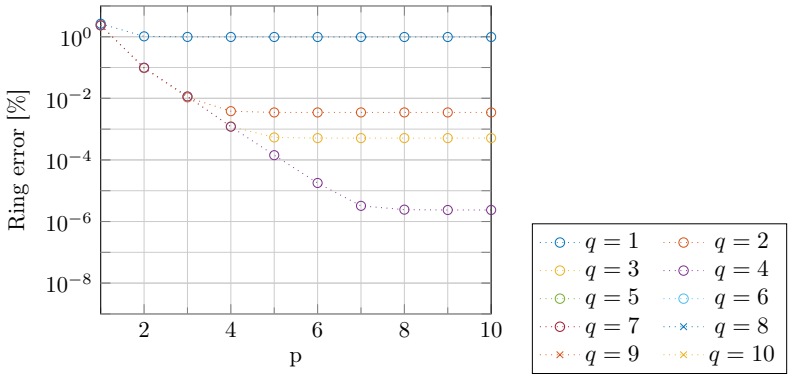
(b) $MpC = 5000$, H^1 modal approach



(c) $MpC = 100$, least squares

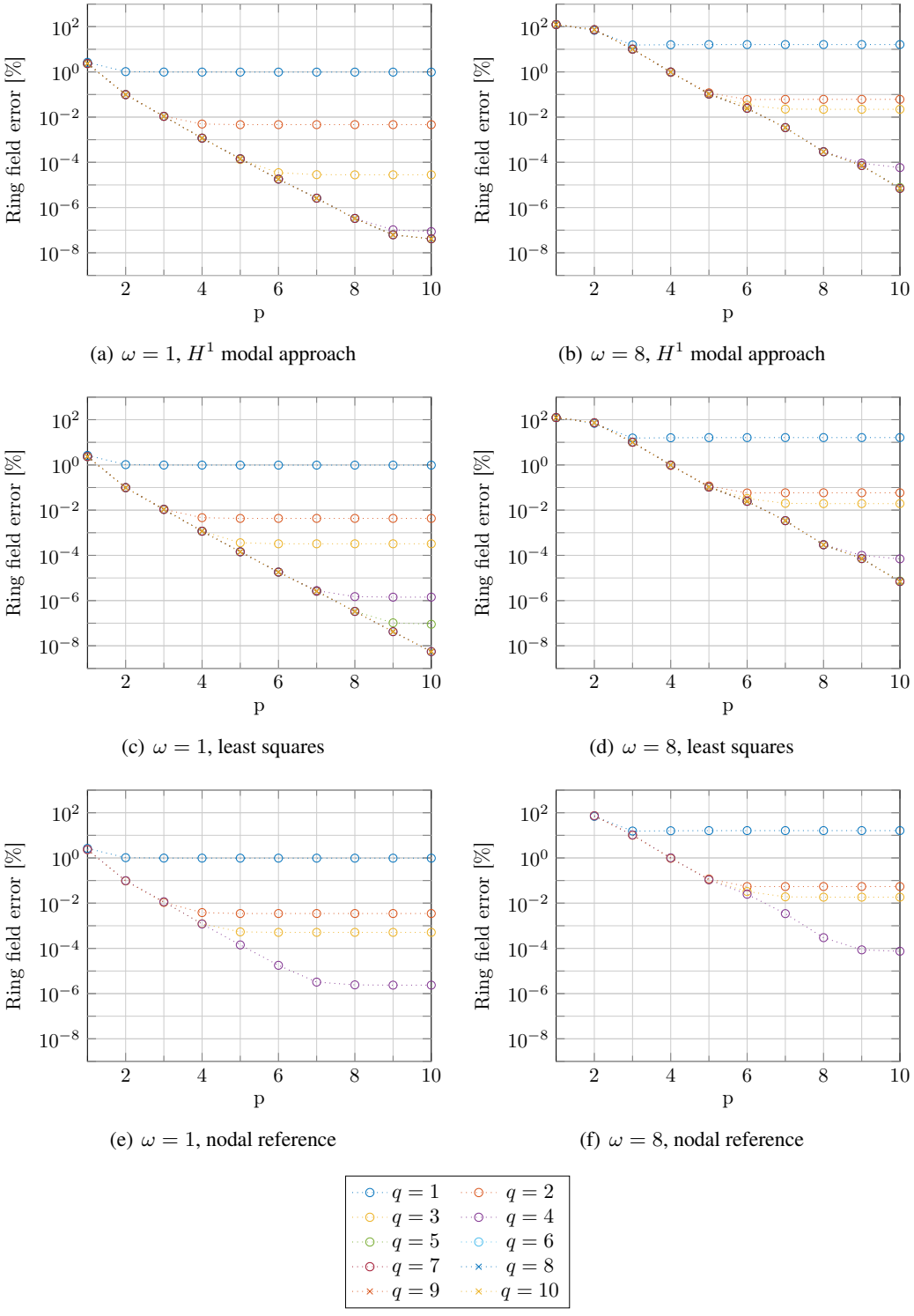


(d) $MpC = 5000$, least squares



(e) nodal reference

Figure 4.5: Comparing p-convergence curves for different MpC-values of the circle scattering test case with 12 Cmesh elements and $\omega = 1$.



(a) $\omega = 1$, H^1 modal approach

(b) $\omega = 8$, H^1 modal approach

(c) $\omega = 1$, least squares

(d) $\omega = 8$, least squares

(e) $\omega = 1$, nodal reference

(f) $\omega = 8$, nodal reference

Figure 4.6: Comparing p-convergence curves for different ω-values of the circle scattering test case with 12 Cmesh elements and $\mathrm{MpC} = 5000$.

domain description. The stagnation value therefore corresponds to the error level induced by the GDE. This is confirmed by Figure 4.4 where it is shown that the stagnation values drop as the number of Cmesh elements is increased and therefore a better representation of the domain geometry is achieved.
For a comparison of different MpC-refinements, Figure 4.5 shows results for $\mathrm{MpC} = 100$ and $\mathrm{MpC} = 5000$. Matching the results of Section 3.3, a high dependency of the H^1 modal approach on the MpC-refinement can be observed. For $\mathrm{MpC} = 100$, higher curving orders do not further improve the corresponding stagnation levels, their curves fall together. In these cases, the stagnation of the GDE due to the inaccuracy of the Mmesh results also in a stagnation of the field error. This is overcome for the higher refinement $\mathrm{MpC} = 5000$, where higher q lead to lower stagnation levels of the simulation error. Also analogously to Section 3.3, no significant dependency on the MpC-refinement of the simulation results on meshes curved with the least squares approach is observed.
These observations hold in general. A specific effect of the geometric inaccuracy on acoustic simulations is shown in Figure 4.6. There, both the Cmesh size and the Mmesh size are fixed and results for simulations with different angular frequencies ω are compared. With increasing frequency, the stagnation levels increase as well. This confirms that high-frequency simulations are more sensitive to geometric inaccuracy than lower frequency simulations.
Figure 4.7 shows the field error for $p = 10$ and a highly refined Mmesh with $\mathrm{MpC} = 5000$, for the two mesh curving methods and the corresponding Gmsh result. The behaviour of the reference and the least squares approach is similar, with slightly better results with the latter. For low ω values, the results obtained on H^1 modally curved meshes are generally better. The development from $q = 3$ to $q = 4$ for $n_{\mathcal{C}}^e = 12$ and from $q = 2$ to $q = 3$ for $n_{\mathcal{C}}^e = 28$ also shows that the ω range increases within which this is the case. As the curving order increases ($q = 5$ for $n_{\mathcal{C}}^e = 12$ and $q = 4$ for $n_{\mathcal{C}}^e = 28$), the field error stagnates on the H^1 modally curved meshes. This is caused by the stagnation of the GDE, which can be related back to the inaccuracy of the Mmesh. Further MpC refinement is expected to overcome the stagnation and to lead to even lower field errors. As the least squares meshes are not effected by the MpC refinement, results on those meshes will not stagnate until the p refinement forms the bottleneck. Higher ω values were dropped as in those cases the p-convergence curves did not stagnate. The resulting error at $p = 10$ does not coincide with the GIE and no conclusions can be drawn from this data about the curving method's effect on the field error.

4.5 Conclusions of this Chapter

In this chapter, the curved boundaries obtained in Chapter 3 are applied to an acoustic simulation. First a short review on basic equations and typical numerical methods in numerical acoustics is provided. This review is extended by a summary of the two mainly used implementations of geometric enhancement in FEM. The embedding of curved boundaries into domain meshes is described and the implementation validated with the acoustic test case of a plane wave scattering by an infinite cylinder, corresponding to a circle in 2D. The results indicate that the accuracy of the simulations strongly depends on the underlying geometric accuracy, with increasing sensitivity to geometric inaccuracy with increasing frequency. Further, the GIE is reduced on

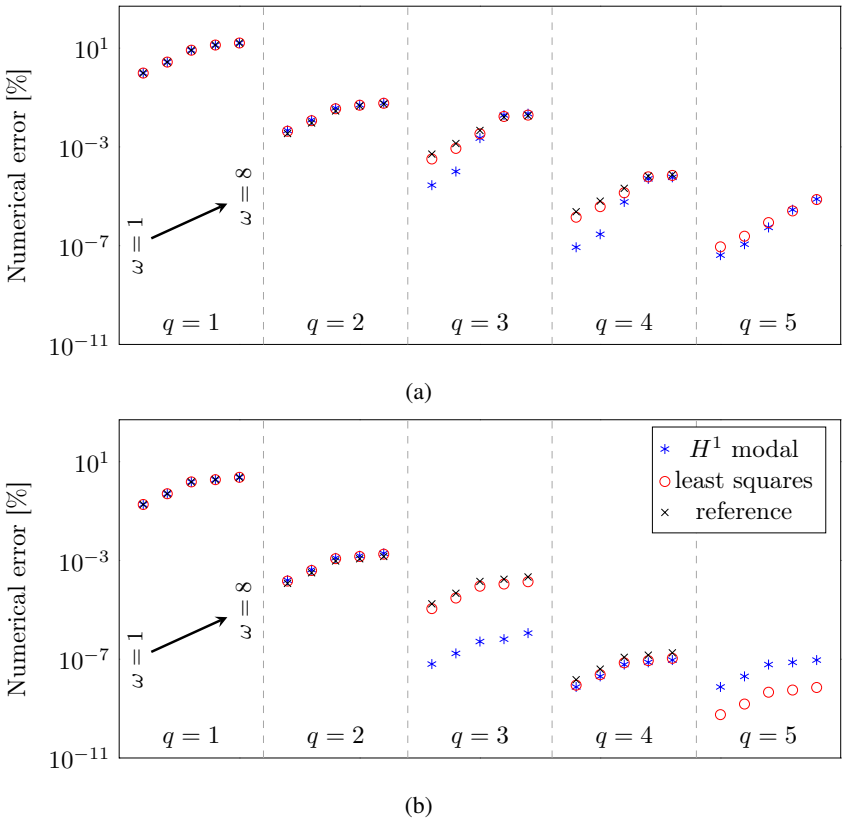


Figure 4.7: Plot of simulation results at $p = 10$ for $q = 1, \ldots, 5$ and $\omega = 1, 2, 4, 6, 8$ for (a) $n_{\mathcal{C}}^e = 12$ and (b) $n_{\mathcal{C}}^e = 28$. The results from the H^1 modal approach are plotted in blue stars, those from the least squares curving in red circles, and the reference is plotted in black plus signs.

meshes curved with the H^1 modal method, providing that the Mmesh refinement is fine enough.

Chapter 5

Relation Between GDE and GIE

As observed in the previous chapter, the choice of both the geometric and the acoustic parameters has an influence on the field error. Figure 5.1 provides an overview over the introduced parameters and error measures and a schematic of their relations. In Chapter 3 the focus lies on the geometry based dependence of the GDE on the selected geometric parameters, while in Chapter 4 some acoustics theory is summarised, including a formula for the expected field error under the assumption that the geometric discretisation is exact. This section considers the missing link: The relation between the geometry discretisation error (GDE) and the geometry induced error (GIE). First, a literature review is given on links that are suggested between the field order p and the geometric order q, as well as on studies on the dependencies of the GIE on the GDE in general. Afterwards, the relationship between the two is investigated in the context of acoustic simulations by applying the qp-FEM to the Helmholtz scattering problem described in the last Chapter.

5.1 Literature Review

5.1.1 Review on Links Between Field Order and Geometric Order

The importance of an accurate domain description in high-order methods is well documented in the literature [79]. In [6], Bassi et al. examine the impact of the geometric description on the performance of a discontinuous Galerkin method (DGM) simulation of flow around obstacles. Isoparametric elements with $p = q$ and superparametric elements with $q = p + 1$ are compared for p up to 3. Generally, linear geometric interpolation leads to inaccurate results, such that the superparametric element with $p = 1$ and $q = 2$ results in a large improvement. However, only small improvements are observed when further increasing p. This indicates that the geometric description should closely follow the field description. The impact of the normal description is also examined and the authors conclude that an accurate description of the normal can often be more important than the geometric description itself.
Contradicting conclusions are drawn in [89], where the effect of the boundary representation is considered on results of DGM simulations for linearised Euler equations. Besides linear and 2^{nd} order elements, also a mixed element type is tested. It consist of a linear representation of the element edges, but the normals of the corresponding quadratic element. The clear advantage of the 2^{nd} order elements in terms of accuracy in the simulation results is not obtained with the

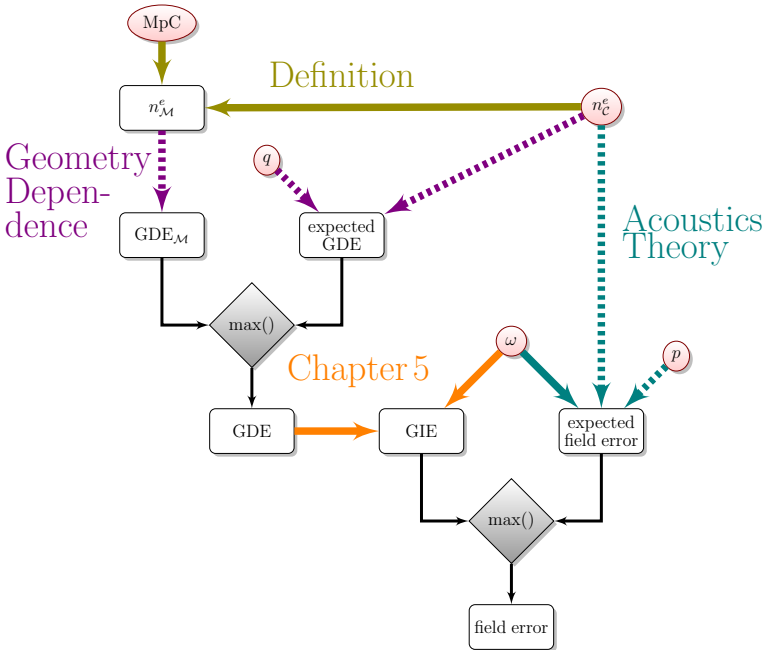


Figure 5.1: Schematic of the error dependence. Input parameter are highlighted with red circles or oval shapes. Coloured solid arrows stand for monotonically increasing functions, dashed arrows for a monotonically decreasing functions. If e.g. the MpC is increased, $n^e_{\mathcal{M}}$ increases as well, while an increase of $n^e_{\mathcal{C}}$ leads to a decrease of the geometry discretisiation error GDE$_{\mathcal{M}}$ of the Mmesh. The colors describe where the explanations of the relations can be found. The geometry dependence was treated in Chapter 3, the acoustics theory in Chapter 4. Black solid arrows indicate the input and output of the $\max()$ functions. The overall GDE is e.g. obtained as the maximum of the GDE$_{\mathcal{M}}$ and the expected GDE that would be obtained with an exact target curve.

mixed element type. The authors conclude that focusing only on the improvement of the normal description is not beneficial in aeroacoustic simulations.
In [5] and shortly afterwards in [22], a theoretical analysis is performed on the convergence rate of the high-order FEM for second-order elliptic boundary value problems. The exponential p-convergence rate is given as

$$\left\| u - u^{(h,p)} \right\|_m \leq \frac{c h^{p+1-m} \left\| u \right\|_r}{p^{r-1}}, \tag{5.1}$$

with u the analytical solution, $u^{(h,p)}$ the approximate solution and $\left\| \cdot \right\|_m$ and $\left\| \cdot \right\|_r$ the m^{th} and r^{th} Sobolev norm, respectively. It is demonstrated that a geometric interpolation of order $q \geq p - 1$ is needed to maintain this convergence rate.
This trend is confirmed by the numerical results given in [54] and [55]. Tension simulations are performed on plates with a circular or elliptic hole, described by elements of geometric order $q \in \{1 \dots 4\}$. As the field interpolation order is increased from $p = 1$ to $p = 10$, the relative error in energy norm exhibits first an exponential convergence, then the convergence rate decreases and finally the error stagnates.

5.1.2 Review on Relation Between GDE and GIE

The error in the geometry of the computational domain boundary corresponds to an inexact problem description, such that the problem actually solved differs from the initial one. By increasing the element order p for a constant geometric description, the field representation becomes more and more accurate. Thereby the solution converges to the analytical results of the wrong problem, and the error level calculated with respect to the analytical solution of the exact problem stagnates (see Figure 5.2). The observed stagnation level of the field error in energy norm corresponds to the GIE. It is important to analyse the dependence between the GIE and the GDE, as depending on the type of problems considered and also depending on the applied numerical method, the geometric description may have a strong or a limited impact on the simulation results.
In [54] and [55], Luo et al. examine the influence of the geometry approximation on results of high-order FEM tension simulations. The effect of the geometric interpolation order q, and of the smoothness of the geometric approximation on the GIE are studied for a circle and an ellipse with a moderate aspect ratio. In both cases, an increase of q leads to a strong decrease of the GIE. In comparison, the use of a C^1-continuous approximation only yields marginal improvements in the error based on the energy norm. Results based on the maximum error are significantly improved using the C^1 boundary description. However, in these two studies, no quantitative measure of the GDE is given.
Another analysis on the stagnation level with respect to geometric accuracy can be found in [91], where a spherical obstacle is discretised with linear or reconstructed quadratic elements. It is remarked that, as the size of the elements describing the sphere decreases, the stagnation level of the error decreases as well. For linear elements, an approximately reciprocal relationship between h^2 and the stagnation level is mentioned. However, this observation is based on a small number of sampling points.

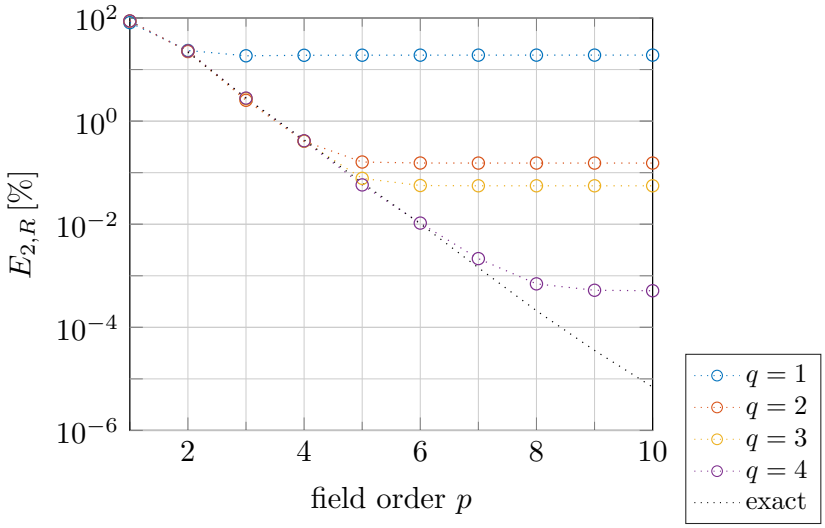


Figure 5.2: Field error stagnation for p-convergence, comparing different q values. The dashed black line indicates the error obtained for simulations with an exact geometric description.

5.2 GDE Measures

This section lists the GDE measures that are used for the studies and how they are evaluated. From the measures described in 2.2, those that are strictly proper, meaning that they are only zero for perfectly matching curves, are considered here. Therefore, the differential geometry based measures are excluded.

All examples within this section are given for the circular scattering boundary of the test case introduced in Section 4.4.1. The high-order representations of the examples were obtained with the Lagrange curving algorithm of the software *Gmsh*, applying also the untangling and smoothing algorithm described in [90]. During the untangling process, boundary nodes are moved. Thereby, only the mesh validity and quality are considered. These are measures that are defined for the mesh to ensure that it contains no tangled elements and that the mesh elements are well shaped. The accuracy of the geometric approximation is neglected. Here, only meshes with $n_{\mathcal{C}}^e = 4$ and $q >= 3$ were tangled and therefore affected by the untangling procedure. In the case of $q = 4$, this leads to an increased geometry discretisation error. For $q = 3$, boundary nodes are only slightly moved in the untangling process. The boundaries of the tangled and the untangled meshes are nearly equal. Thus, the tangled mesh and the untangled mesh of order $q = 3$ have about the same accuracy.

Hausdorff and Fréchet distances

The evaluation of the Hausdorff distance GDE_H (see (2.7), Section 2.2.1) is based on two nested optimisation problems:

The inner optimisation takes a given point P_i on the approximating curve $\mathcal{G}_{\text{approx}}$ and finds the point $Q_{i,\text{min}}$ on the exact curve $\mathcal{G}_{\text{exact}}$ with the minimal distance $d_{\text{min}}(P_i)$ to P_i. To obtain symmetry, a minimisation is run for points Q_i on $\mathcal{G}_{\text{exact}}$ to find the point $P_{i,\text{min}}$ on $\mathcal{G}_{\text{approx}}$ with the minimal distance $d_{\text{min}}(Q_i)$ to Q_i. The outer optimisation maximises $d_{\text{min}}(P_i)$ with respect to P_i by varying the reference point on the reference interval. Analogously $d_{\text{min}}(Q_i)$ is maximised with

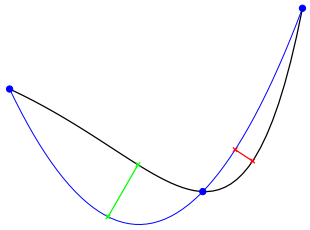
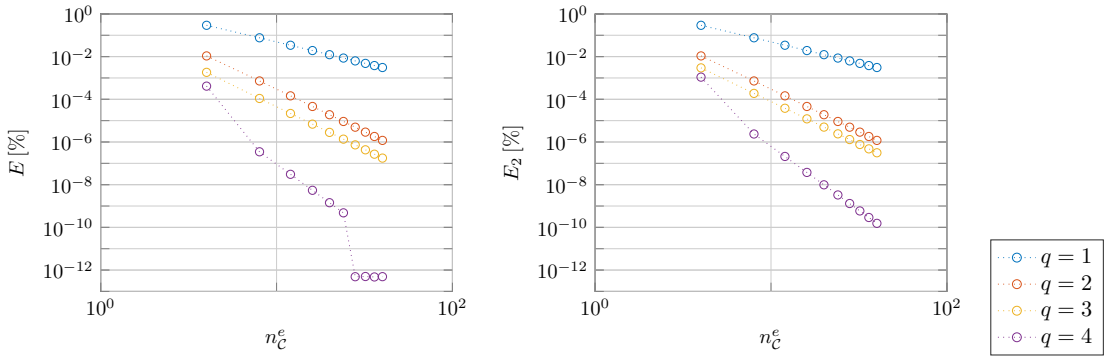


Figure 5.3: Schematic of a (red) local maximum that could be found in the elementwise Hausdorff distance calculation. In green, the global maximum distance is shown. With a segmentwise Hausdorff distance calculation, the global maximum is found.



(a) (b)

Figure 5.4: Plots comparing Hausdorff distances evaluated (a) elementwise and (b) segmentwise for the unit circle.

respect to Q_i by varying the parametrising angle.

When the optimisation is computed elementwise, the outer optimisation could run into local maxima, as it is illustrated in Figure 5.3. In order to avoid this, the optimisation is restricted on segments between two nodes. In Figure 5.4, plots of results for both an elementwise and a segmentwise evaluation are displayed. The comparison shows that the elementwise evaluation leads to smaller values than segmentwise evaluation. This is especially visible for $q = 4$. Thus only a local maximum is found during elementwise optimisation. As the tested geometry is a circle, the segmentwise evaluation is assumed to provide the global optima and is the method that will be applied from here on.

To constrain the maximisation on the given element or segment S, $d_{\text{min}}(P_i)$ is set to zero for $P_i \notin S$, and analogously $d_{\text{min}}(Q_i) = 0$ for $Q_i \notin S$.

As mentioned in Section 2.2, the Hausdorff distance and the Fréchet distance are equal for closed, convex curves. For convex target curves such as the circle and the ellipse, this can be fulfilled for $q \in \{1, 2\}$. But for cubic and quartic geometric order, the interpolation oscillates around $\mathcal{G}_{\text{exact}}$, which leads to non-convex curves. Therefore, the Fréchet distance and the Hausdorff distance are compared.

For the Fréchet distance, the algorithm given in [59] is implemented, and, for comparison, a discrete Hausdorff distance. Therefore both curves, $\mathcal{G}_{\text{exact}}$ and $\mathcal{G}_{\text{approx}}$, are evaluated on n_{ref} points p_i. These points are equidistant in ξ for the approximating curve and equidistant in the angle θ for the exact curve. The distance is computed separately for each element and then the maximum

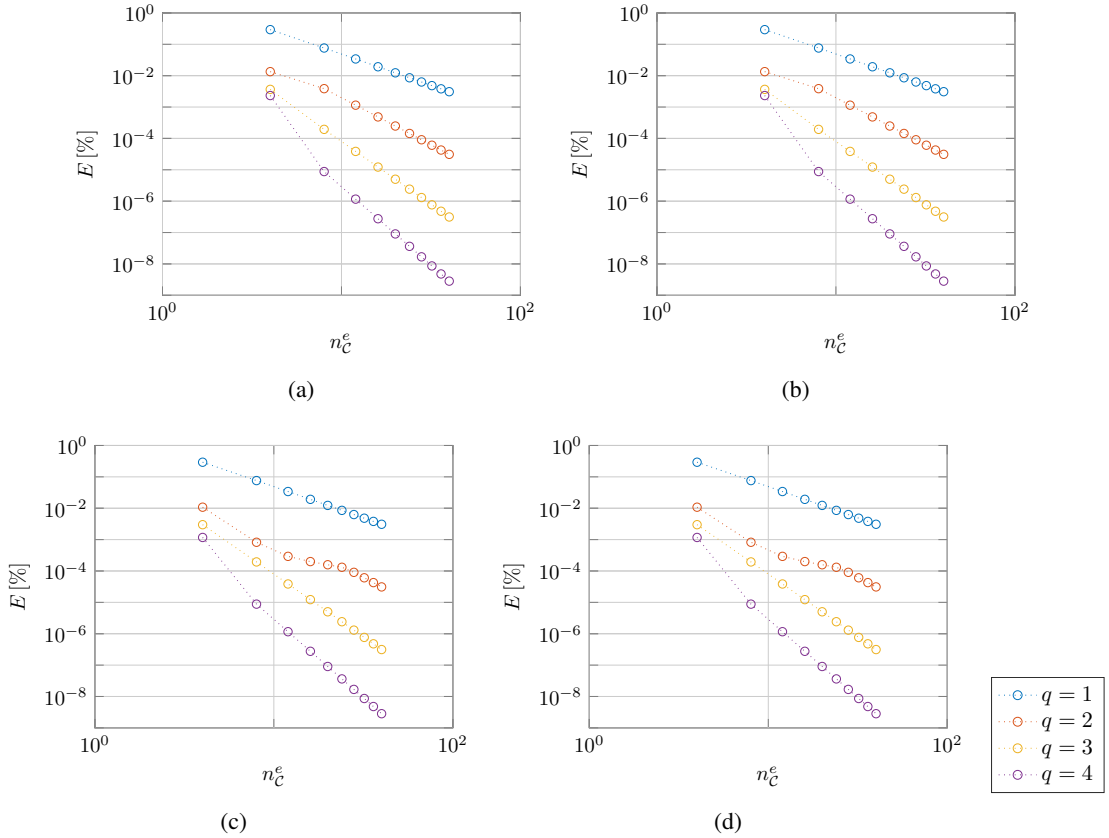


Figure 5.5: Plots comparing Hausdorff (left) and Fréchet distances (right) evaluated with (a), (c) 100 refinement points and (b), (d) 1000 refinement points for the circular scattering boundary.

is taken over all elements.

In Figure 5.5 results for the discrete Fréchet and Hausdorff distance for two different refinements of the circular scattering boundary can be seen.Within machine precision, both measures provide the same distance values. Especially for $q = 2$, some deviation from the results for the optimisation-based Hausdorff distance (see Fig. 5.4(b)) can be seen as well. These deviations decrease notably for the meshes with smaller $n_{\mathcal{C}}^e$ as the number of refinement points n_{ref} is increased. Considering this, and the limit on the number of refinements due to high computational costs, it is decided to drop the Fréchet distance as a measure of the GDE.

Area based measure

As a more intuitive measure, the area based GDE measure (3.4) introduced in Section 3.3.1.2 is computed. In Figure 5.6, GDE_A is shown for the circle example geometry.

Domain mapping based measure

The mapping based GDE measure introduced in [97] has to be adjusted for our 2D test case. In order to allow an elementwise integration, $\boldsymbol{d}_\Omega(x)$ is transformed to the reference space and is calculated there as the difference between the exact and the approximate geometric mapping. For the example geometry, the exact mapping of $\hat{T}$ to a T in the mesh is obtained using a linear Lagrange mapping in polar coordinates (r, θ):

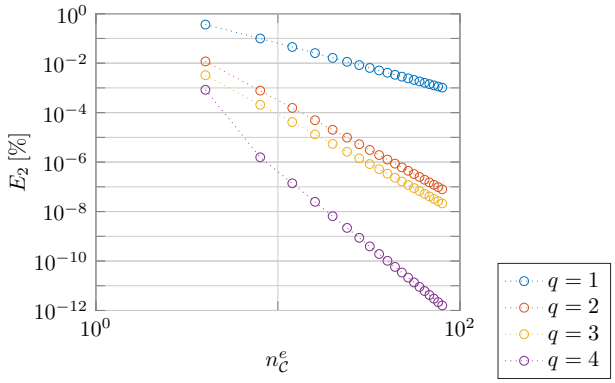


Figure 5.6: Plot showing the relative area GDE_A between the circle and its approximating surface.

$$
\begin{aligned}
x(\xi,\eta) = &\left(\left(\frac{r_2 - r_1}{2}\right)\xi + \left(\frac{r_3 - r_1}{2}\right)\eta + \frac{r_2 + r_3}{2}\right) \cdot \\
&\cos\left(\left(\frac{\theta_2 - \theta_1}{2}\right)\xi + \left(\frac{\theta_3 - \theta_1}{2}\right)\eta + \frac{\theta_2 + \theta_3}{2}\right) \\
y(\xi,\eta) = &\left(\left(\frac{r_2 - r_1}{2}\right)\xi + \left(\frac{r_3 - r_1}{2}\right)\eta + \frac{r_2 + r_3}{2}\right) \cdot \\
&\sin\left(\left(\frac{\theta_2 - \theta_1}{2}\right)\xi + \left(\frac{\theta_3 - \theta_1}{2}\right)\eta + \frac{\theta_2 + \theta_3}{2}\right),
\end{aligned}
$$

where (r_i, θ_i) are the polar coordinates of the i^{th} node of T.

The calculation of the $H^1(\Omega)$ seminorm as described in [97] does not only consider the mapping of the boundary line elements but also evaluates the mapping of the domain interior elements. For the here considered meshes, the interior elements are in general kept straight sided with respect to Cartesian coordinates (beside the case for $n_{\mathcal{C}}^e = 4$). They are especially not optimised to follow the linear mapping in polar coordinates.

In Figure 5.7, plots of a measure evaluated over the whole domain and one evaluated only over the curved domain boundary are shown. Note that instead of the $H^1(\Omega)$ seminorm, the $L^2(\Omega)$ norm is computed in both cases. The GDE based on integration over the whole physical domain is dominated by the influence of interior elements.

For this thesis, the GDE measures are meant to measure the geometric accuracy of the approximation of the boundary. Thus, the domain based GDE calculated with the $L^2(\Omega)$ norm is dropped. Only the boundary based GDE calculated with the $L^2(\Gamma_s)$ norm is kept and called GDE_M.

5.3 Field Error Measures

As described in Section 4.4.2, the standard field error evaluation procedure has to be replaced by alternative approaches. Besides the evaluation along the ring that was described and used in the previous chapter, two additional methods are presented here and tested for the circle scattering test case:

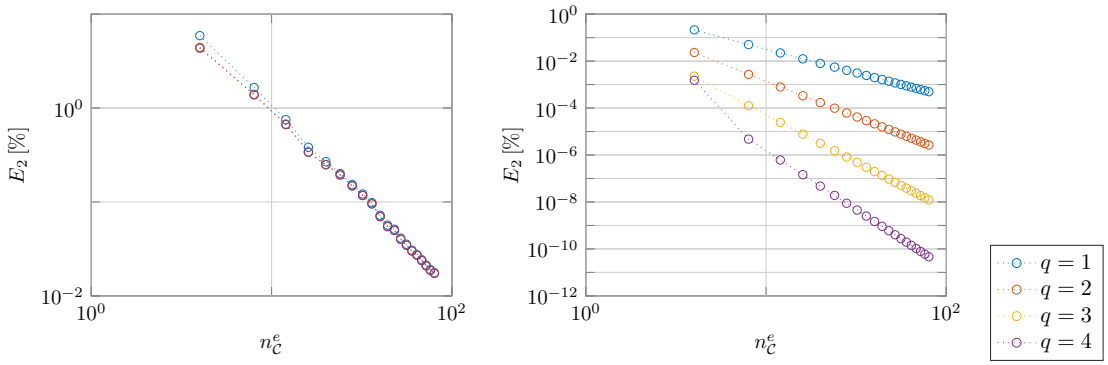


Figure 5.7: Geometry discretisation error (GDE) based on [97], using the L^2-norm over (a) the physical domain and (b) the boundary describing the scatterer.

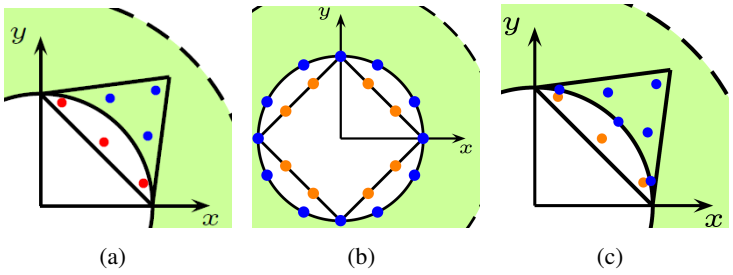


Figure 5.8: Schematic figures showing (a) the problem in the standard field error evaluation for the scattering test case and (b)-(c) the two additional alternative approaches. The red dots in (a) are the quadrature points that fell out of Ω (green). In (b) and (c), blue dots mark the quadrature points that automatically fall inside of Ω, and those points on which the analytical solution is evaluated in the cases when quadrature points fall outside of Ω.

- **Approach based on domain mapping** (see Figure 5.8(b))
  In [97] an error evaluation method based on the exact mapping is defined. Analogously to the GDE_M described in Section 2.2, the exact solution at a given point $\mathrm{p}_{\mathrm{ex}}(\boldsymbol{x})$ in the physical domain is not compared to the simulation result at this point. Instead, it is compared to the simulation results $\mathrm{p}\left(\hat{\boldsymbol{x}}\right) = \mathrm{p}\left(\boldsymbol{\zeta}\left(\boldsymbol{x}\right)\right)$, evaluated at the point $\boldsymbol{\zeta}\left(\boldsymbol{x}\right)$ that approximates $\boldsymbol{x}$ in the discretised geometry (comp. Section 2.2.3). For the field error, the H^1-seminorm of $\mathrm{p}_{\mathrm{ex}}(\boldsymbol{x}) - \mathrm{p}\left(\boldsymbol{\zeta}\left(\boldsymbol{x}\right)\right)$ is computed and divided by the H^1-seminorm of $\mathrm{p}_{\mathrm{ex}}(\boldsymbol{x})$.

  Following the arguments for the GDE measures, the field error measure based on the H^1-seminorm is adapted by considering the $L^2(\Gamma_s)$ norm. Thus, the relative error is evaluated as

$$
\begin{aligned}
E_{2,M} &:= \sqrt{\frac{\int_{\Gamma_s} \left|\mathrm{p}_{\mathrm{ex}}\left(\boldsymbol{x}\right) - \mathrm{p}\left(\boldsymbol{\zeta}\left(\boldsymbol{x}\right)\right)\right|^2 \,\mathrm{d}\Gamma_s}{\int_{\Gamma_s} \left|\mathrm{p}_{\mathrm{ex}}\left(\boldsymbol{x}\right)\right|^2 \,\mathrm{d}\Gamma_s}} \\
&\approx \sqrt{\frac{\sum_i w_i J_{\mathrm{ex}} \left|\mathrm{p}_{\mathrm{ex}}\left(\boldsymbol{x}\left(\boldsymbol{\xi}_i\right)\right) - \mathrm{p}\left(\hat{\boldsymbol{x}}\left(\boldsymbol{\xi}_i\right)\right)\right|^2}{\sum_i w_i J_{\mathrm{ex}} \left|\mathrm{p}_{\mathrm{ex}}\left(\boldsymbol{x}\left(\boldsymbol{\xi}_i\right)\right)\right|^2}},
\end{aligned}
$$

  where J_{ex} is the Jacobian of the exact mapping and w_i the weights of the Gauss quadrature.

- **Projecting method** (see Figure 5.8(c))
  The third method that is considered is based on an integration over the whole domain. Using Gaussian quadrature on $\hat{T}$, the L^2-error $E_{2,P}$ is evaluated elementwise. For quadrature points whose corresponding points $\hat{\boldsymbol{x}}_i$ in T lie inside the scatterer, the analytical solution is evaluated at the projection of $\hat{\boldsymbol{x}}_i$ along radial direction on the scattering domain. With this approach, an evaluation of the analytical solution outside of its domain of definition is prevented.

To compare the described measures, the scattering by the circle (see Section 4.4.1) is considered. Polynomial higher-order meshes that were curved using the software *Gmsh* (see Section 3.1) are used. For the field interpolation, Lobatto shape functions are applied. Using qp-FEM elements, the geometric order q and the field order p can be chosen separately. Figure 5.9 provides a comparison between the three different error evaluation methods for the set of simulations with $\omega = 1$ and $h \approx 0.79$.

All three methods show the expected general behaviour: As p is increased, the field error first converges exponentially, then stagnates at a certain level. As the error induced by the field interpolation is continually decreased by increasing p, the level of stagnation corresponds to the GIE. According to [5] and [22], the stagnation is expected to begin at $p^* = q + 1$. The corresponding points are marked in the plot with a black dot and seem, besides for $q = 1$ to fall together with the onset of the stagnation. In conformity with the literature ([54], [55]), the stagnation level decreases with increasing q.

Besides this accordance in the general behaviour, the level at which the error stagnates varies with the evaluation method. The field error evaluated with the projecting method and the one

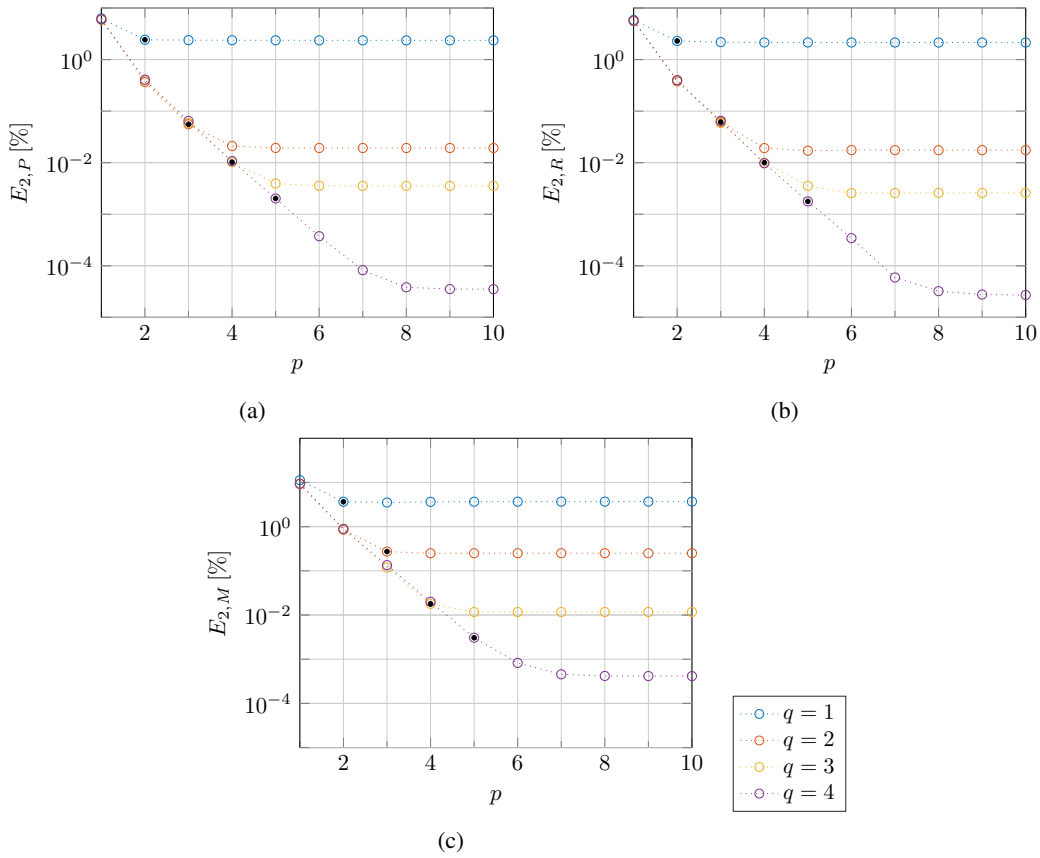


Figure 5.9: Stagnating p convergence curves of the field error for simulations with $\omega = 1$ and $h = \frac{2\pi}{8} \approx 0.79$, evaluated (a) with the projection based method, (b) along the ring $r = 1.5$, and (c) with the mapping based method. The black dot indicates the expected onset of stagnation $p^* = q + 1$.

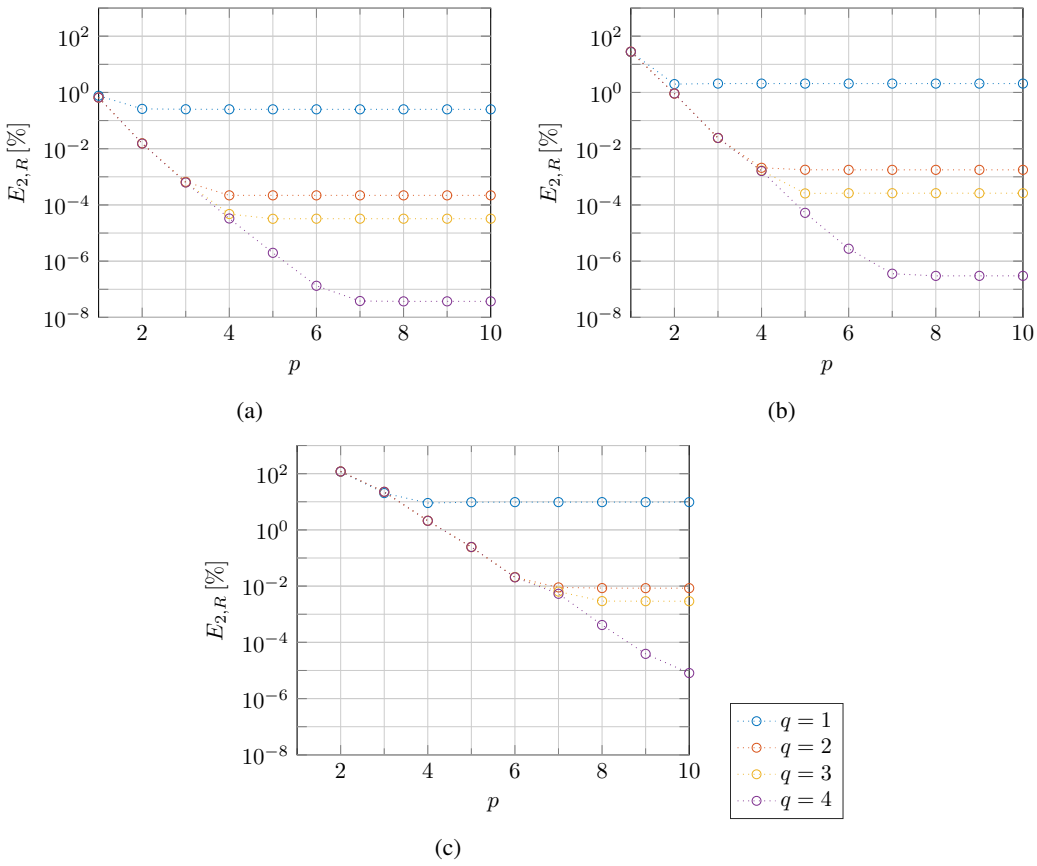


Figure 5.10: Stagnating p convergence curves of the field error evaluated along the ring $r = 1.5$ for a fixed mesh with $h = \frac{2\pi}{24} \approx 0.26$ and (a) $\omega = 1$, (b) $\omega = 4$, and (c) $\omega = 16$.

evaluated along the ring show a similar behaviour. Their stagnation level of the second and third order lying closer together in logarithmic scale than the levels of first and second or third and fourth order. This follows the trend seen in GDE$_A$ (see Fig. 5.6) and GDE$_H$ (see Fig. 5.4). On the other hand, the stagnation levels for the mapping based error evaluation is more consistent with GDE$_M$ (see Fig. 5.7).

5.4 Finding a Model of Relation between GDE and GIE

With the evaluation of the geometric and the field error in place, the GDE-GIE relation can now be studied. Again, the scattering of a plane wave by a cylinder is considered on meshes that were nodally curved with *Gmsh*. To relate the GIE to the GDE, the previously discussed stagnation level is evaluated for simulations with frequencies $\omega \in \{1, 2, 4, 6, 8, \ldots, 28, 30, 32, 64\}$ and mesh sizes $n_C^e \in \{4, 8, 12, \ldots 72, 76, 80\}$. This stagnation level corresponds to the GIE and is denoted $\tilde{E}_{2,M}$, $\tilde{E}_{2,R}$ and $\tilde{E}_{2,P}$, where the subindices M, R, and P refer to the mapping based, ring based, and projection based field error evaluation method, respectively. Simulation results that did not reach a plateau are not considered in the following evaluations.

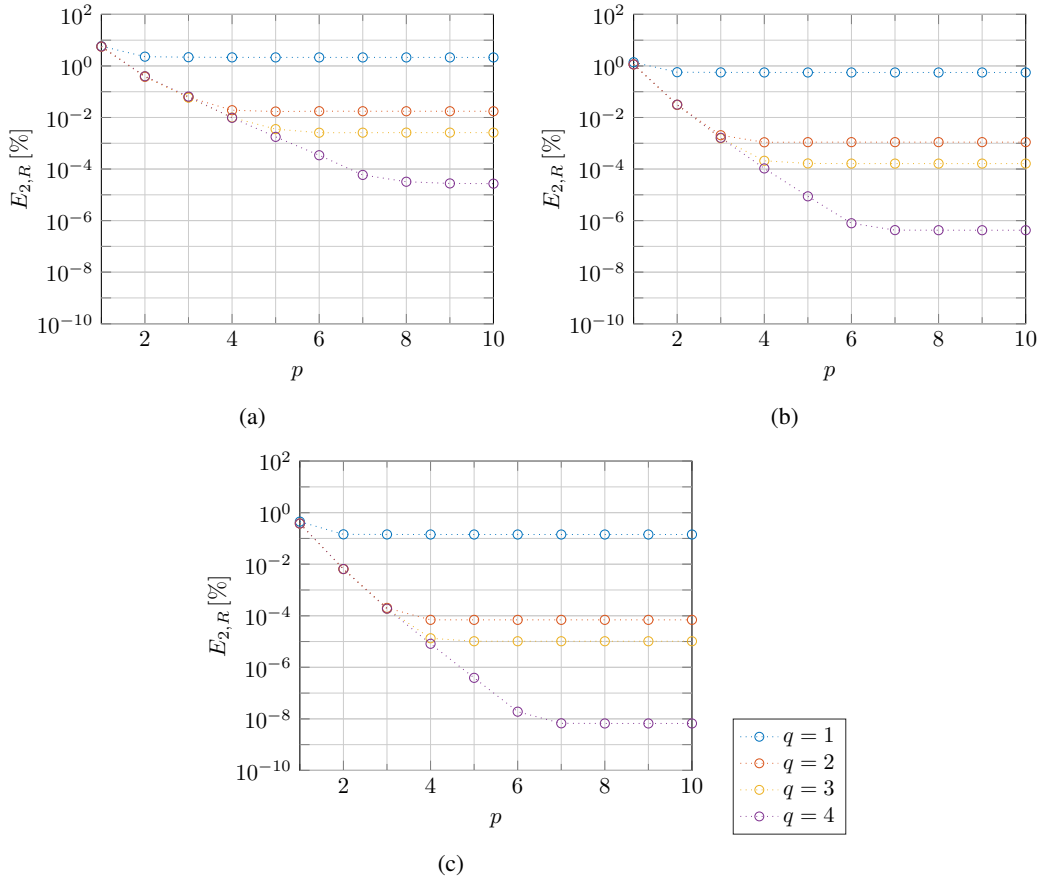


Figure 5.11: Stagnating p convergence curves of the field error evaluated along the ring $r = 1.5$ for a fixed $\omega = 1$ and varying mesh size (a) $h = \frac{2\pi}{8} \approx 0.79$, (b) $h = \frac{2\pi}{16} \approx 0.39$, and (c) $h = \frac{2\pi}{32} \approx 0.20$.

5.4.1 General Dependence on ω and n_C^e

Examples for ring evaluations are given in Figures 5.10 and 5.11, to show the influence of ω and the number n_C^e of line elements along the circular scattering boundary Γ_s on the stagnation levels. Increasing the frequency while keeping the meshes fixed leads also to an increase of the geometry stagnation level. Thus, the higher the frequency, the more sensitive is the field computation to the GDE. As expected, applying an h refinement by increasing the number of elements describing the scattering boundary leads to a decrease of the GIE, since the geometry is better resolved. These two general effects can be observed for all three field error evaluation methods.

5.4.2 Model Building

First tests show that the mapping based evaluated error $\tilde{E}_{2,M}$ and the mapping based GDE_M are related to each other, but to none of the other GDEs or field errors. This is expected from the comparison of the plots of the different GDEs and error evaluation methods. Therefore, only the combinations indicated in Table 5.1 are considered in the following. The corresponding data sets can be seen in Figure 5.12.
A linear dependency can be observed between $\log_{10}(\tilde{E}_2)$ and $\log_{10}(\text{GDE})$ for fixed ω. Thereby, no or only a marginal dependency on q is present. The intercept of the linear relation is changed by an offset that increases with increasing frequencies. A logarithmic dependency between this

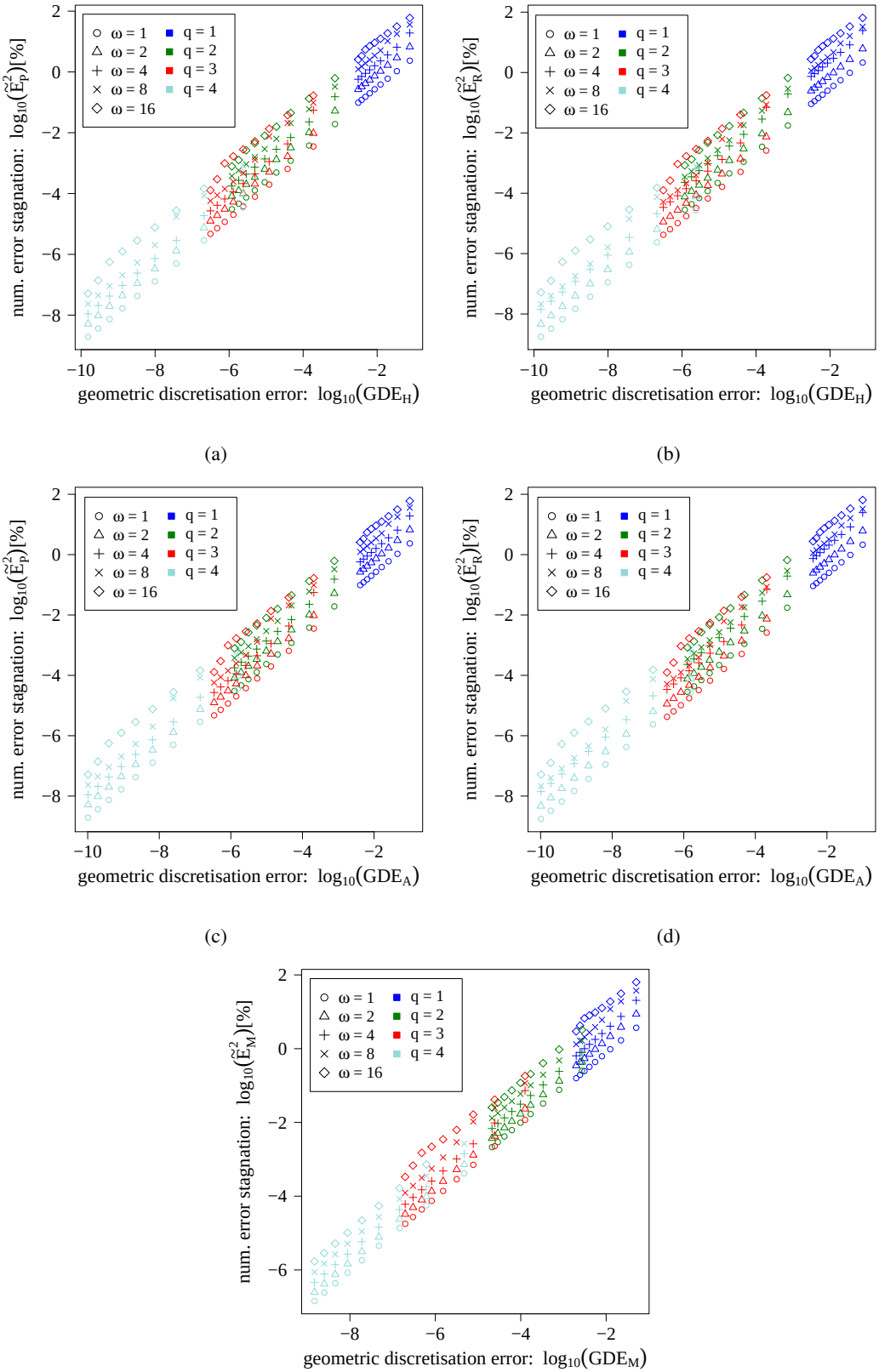


Figure 5.12: Plots of the data sets used for fitting the GIE $\tilde{E}_2$ against the GDE. They show (a) $\tilde{E}_{2,P}$ against GDE_H, (b) $\tilde{E}_{2,R}$ against GDE_H, (c) $\tilde{E}_{2,P}$ against GDE_A, (d) $\tilde{E}_{2,R}$ against GDE_A, and (e) $\tilde{E}_{2,M}$ against GDE_M. Only a subset of the full data set is shown for clarity.

	GDE_M	GDE_A	GDE_H
$\tilde{E}_{2,M}$	X	-	-
$\tilde{E}_{2,R}$	-	X	X
$\tilde{E}_{2,P}$	-	X	X

Table 5.1: Combinations of GDEs and differently evaluated field errors that are considered to study the relation between the GDE and the GIE.

offset and ω is suggested by Figure 5.12. Combining these observations, the suggested model is of the form

$$\log_{10}(\tilde{E}_2) \sim \beta_1 + \beta_2 \log_{10}(\text{GDE}) + \beta_3 \log_{10}(\omega). \tag{5.2}$$

A linear model fit is performed with the statistical software *R* [69]. Rearranging the model results in the non-linear form

$$\tilde{E}_2 \sim 10^{\beta_1} \text{GDE}^{\beta_2} \omega^{\beta_3}. \tag{5.3}$$

The parameters β_i of the different models are given in Table 5.2. Remarkable is the approximately linear dependency on the geometric accuracy for all models, as well as the sup-linear dependency on the frequency for the models without the mapping based measures GDE_M and $\tilde{E}_{2,M}$. Thinking of the 1D example of a plane wave reflected at a hard wall, geometric inaccuracy consists of a shift in the positioning of the wall. This results in a phase shift and the corresponding field error is expected to grow linearly with the geometric inaccuracy. In the present test case, the sup-linear dependency is assumed to arise from the more complex scattering phenomena in 2D, such as creeping waves along the scatterer or scattering by the normal discontinuities at element vertices.

Note that, as β_1 represents the general error level, it is expected to be problem dependent.

In Table 5.2, also the values of the adjusted $\mathtt{R}^2$ are provided. It measures the portion of the variance in the data that is covered by the model, adjusted to the number of data and the number of explanatory variables β_i. The closer the $\mathtt{R}^2$ is to 1, the better is the model fit.

The definition of the $\mathtt{R}^2$ is based on the general requirements of the linear model, namely that the residuals are uncorrelated and identically distributed. To ensure this and therefore the validity of the $\mathtt{R}^2$, plots of the residuals vs. fitted values are considered (see Fig. 5.13). Strong patterns indicate correlated residuals and would suggest that the model is not suitable. For identically distributed residuals, the distribution of the residuals around zero should not vary along the x-axis. As expected for realistic data, the residuals do not perfectly fulfil these requirements. But especially for those model fits with the field error evaluation along the ring, they look uncorrelated and equally distributed enough to proceed with the considered model.

Plots of the fits are shown in Figure 5.14. All five models provide an excellent fit. For the model relating $\tilde{E}_{2,M}$ and GDE_M, this is presumably induced by the similarity in the definitions of the mapping based error evaluation $E_{2,M}$ and the mapping based GDE_M. In general, these two measures remain impractical as they need to be adjusted to each individual geometry.

Even though for the circle test case GDE_A and GDE_H are nearly equal (see Figure 5.15(a)), this will not necessarily be the case for other geometries. Figure 5.15(b) shows a schematic example for that, where a bump in $\mathcal{G}_{\text{exact}}$ was not resolved by $\mathcal{G}_{\text{approx}}$. The GDE_H would be given by the distance from the tip of the bump to $\mathcal{G}_{\text{approx}}$, which would result in a GDE_H that is much bigger

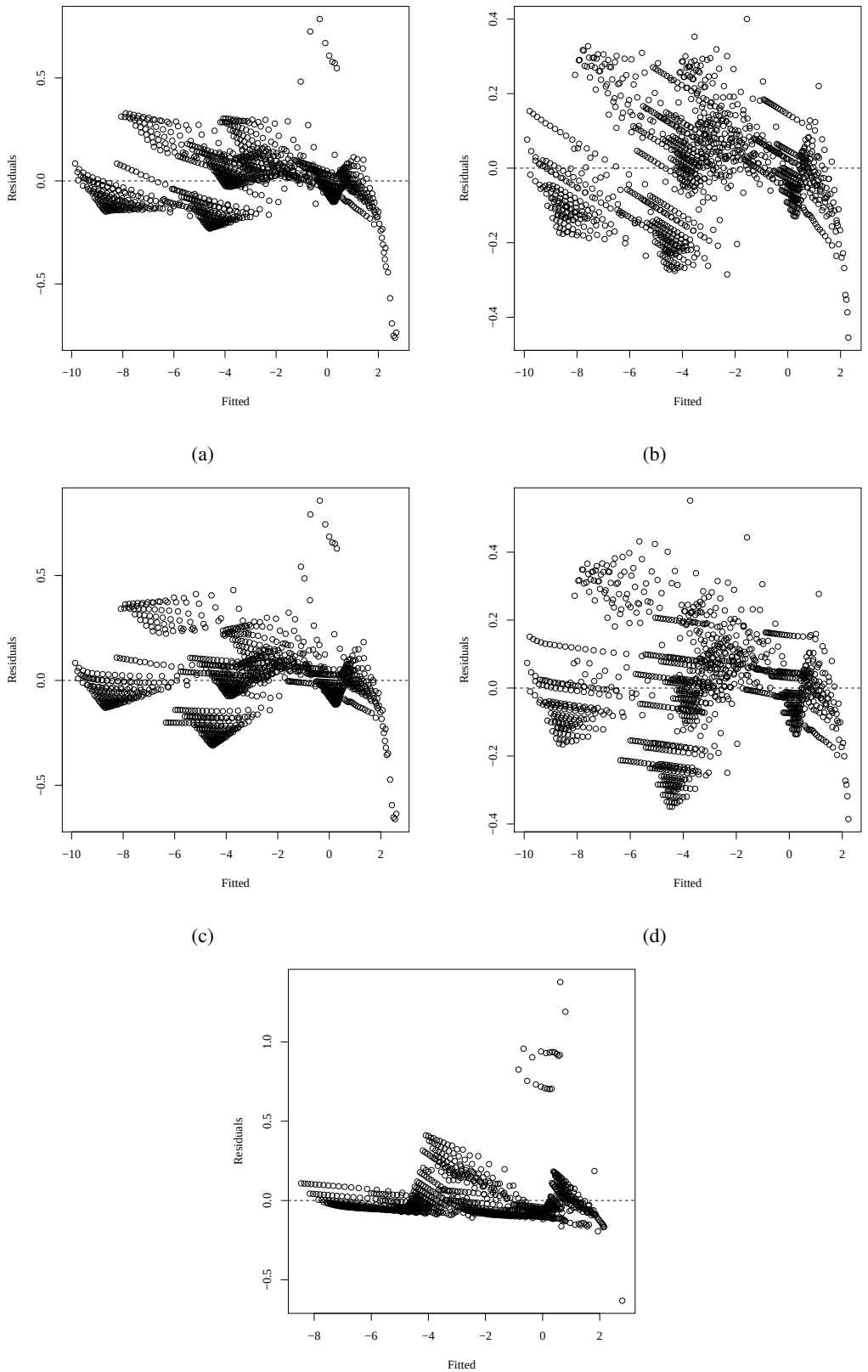


Figure 5.13: Plots of the residuals vs. the fitted data with model (5.2) with (a) $\tilde{E}_{2,P}$ against GDE_H, (b) $\tilde{E}_{2,R}$ against GDE_H, (c) $\tilde{E}_{2,P}$ against GDE_A, (d) $\tilde{E}_{2,R}$ against GDE_A, and (e) $\tilde{E}_{2,M}$ against GDE_M.

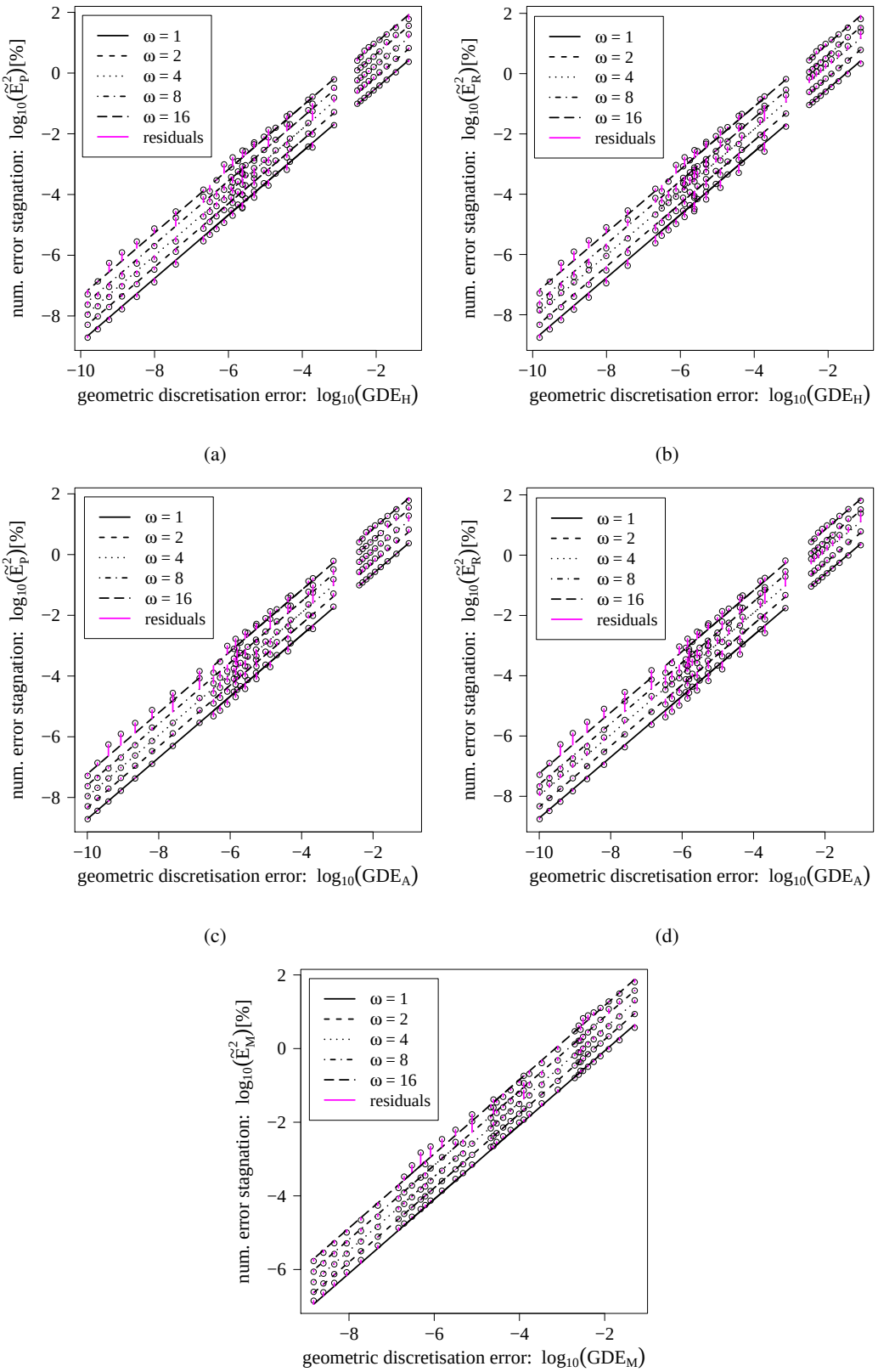
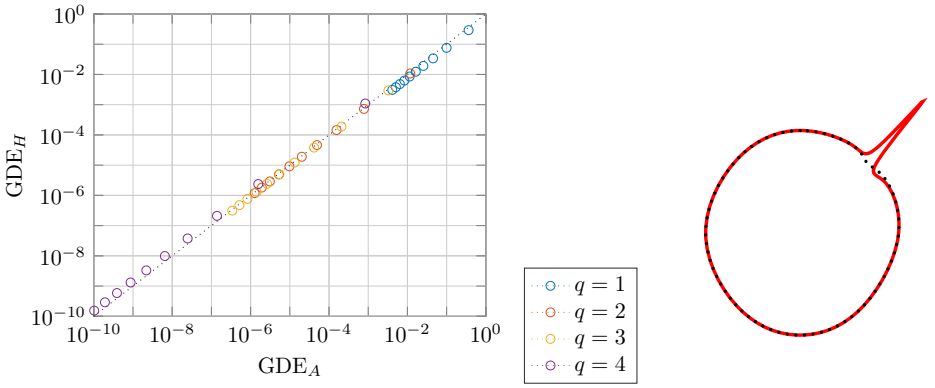


Figure 5.14: Plots of the fitting of the GIE $\tilde{E}_2$ against the GDE with model (5.2). They show (a) $\tilde{E}_{2,P}$ against GDE$_H$, (b) $\tilde{E}_{2,R}$ against GDE$_H$, (c) $\tilde{E}_{2,P}$ against GDE$_A$, (d) $\tilde{E}_{2,R}$ against GDE$_A$, and (e) $\tilde{E}_{2,M}$ against GDE$_M$. Only a subset of the full data set is shown for clarity.

	β_1	β_2	β_3	$\tilde{\mathtt{R}}^2$
$\tilde{E}^2_M, GDE_M$	2.0054	1.0122	1.0105	0.9968
$\tilde{E}^2_R, GDE_A$	1.3846	1.0106	1.2314	0.9975
$\tilde{E}^2_R, GDE_H$	1.5913	1.0456	1.2445	0.9981
$\tilde{E}^2_P, GDE_A$	1.4117	1.0104	1.2159	0.9973
$\tilde{E}^2_P, GDE_H$	1.5990	1.0437	1.2345	0.9978

Table 5.2: Parameter values of the fits of model (5.2) to the indicated combinations of field error evaluation and GDE definitions, and the corresponding values of the adjusted $\mathtt{R}^2$.



(a) (b)

Figure 5.15: (a) shows that for the circle case, the Hausdorff distance GDE$_H$ and the area based GDE$_A$ are very similar. The black line indicates the case when both would be identical. In (b), the red $\mathcal{G}_{\text{exact}}$ is approximated by the black dotted curve. This provides an example to show that GDE$_H$ and GDE$_A$ are generally not identical.

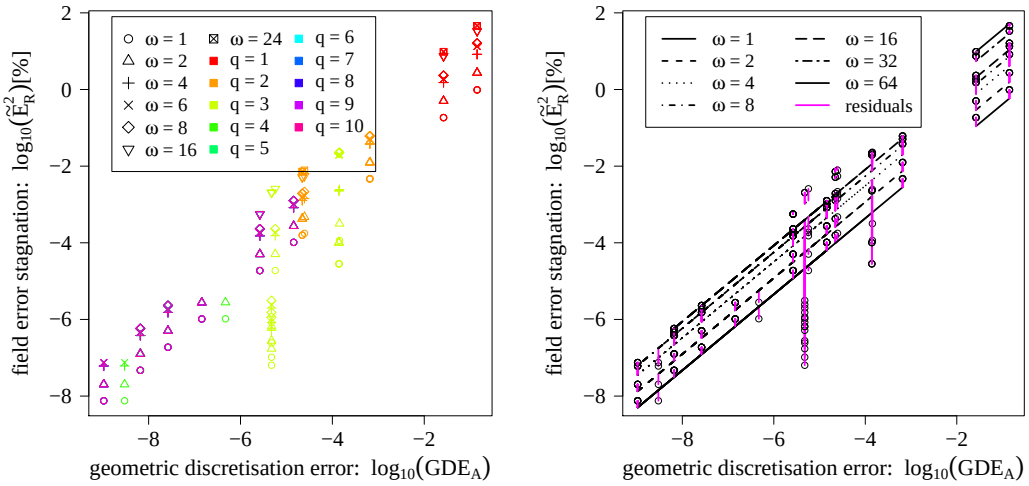
than in the circle test case. On the other hand, since the bump is thin, the GDE$_A$ should not have increased much compared to the circle case.
For the evaluation of the field error it is decided to drop the projecting method. It leads to a qualitatively similar behaviour as the evaluation along the ring, but introduces an extra error in the integration due to the shifted evaluation of the integrand that is hard to control.

5.5 Model Testing with Data from Modally Curved Meshes

In the previous section, only simulations on nodally curved meshes that were created with the software *Gmsh* were considered to evaluate the GDE-GIE-relationship. In this section, the identified model is tested against more data sets that are obtained on meshes that were modally curved with the H^1 modal approach (see Sec. 3.2.2.2) or the least squares approach (see Sec. 3.2.2.1). First, the test case remains the scattering by the cylinder in 2D. In a second step, also the scattering by the distorted ellipse (see Section 3.3.2) is considered, to test the relation model with a more complicated geometry.
The content of this section is part of the paper [99].

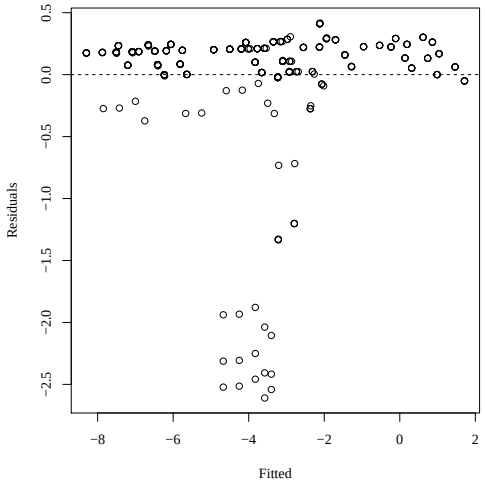
5.5.1 Scattering of a Plane Wave by a Cylinder (2D), Modally Curved Meshes

In Figure 5.16, the full data set is presented for the H^1 modal curving, as well as the corresponding fit with model (5.2) and the residuals vs. fitted plot. In Figure 5.17, the



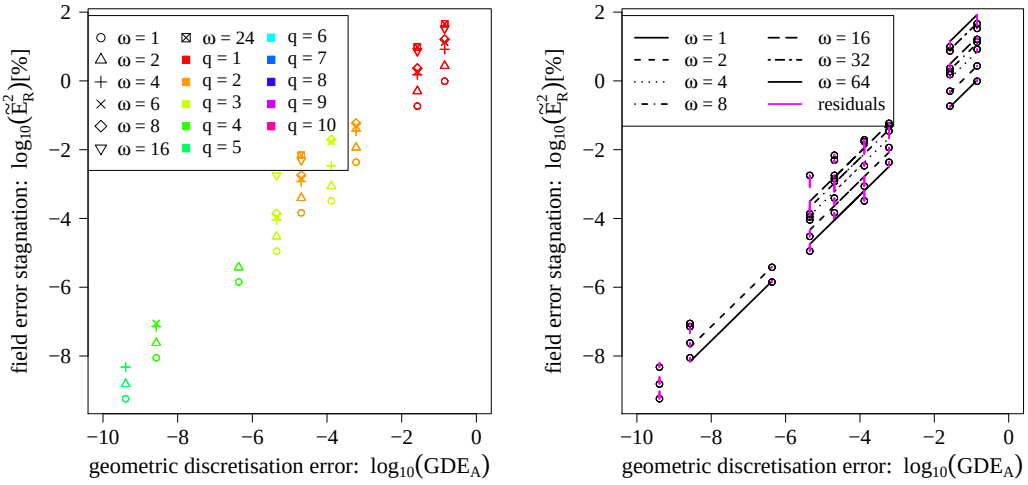
(a) data, all q

(b) fitted, all q



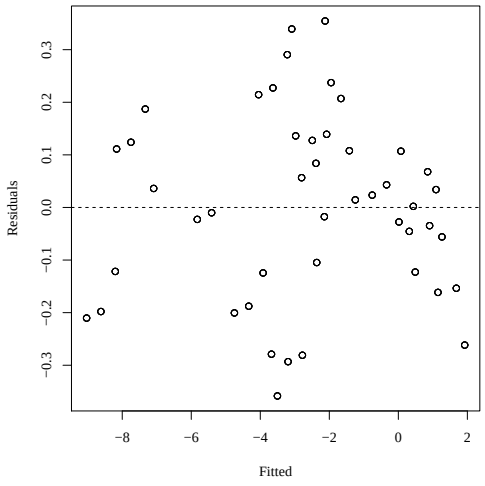
(c) residuals vs. fitted, all q

Figure 5.16: Plots of the fitting of the GIE $\tilde{E}_R^2$ against the GDE for the H^1 modal curving with model (5.2) and the corresponding residuals vs. fitted plot. In (a), the colour decodes the geometric order of the mesh, while the shapes decode the ω value of the simulations.



(a) data, all q

(b) fitted, all q



(c) residuals vs. fitted, all q

Figure 5.17: Plots of the fitting of the GIE $\tilde{E}_R^2$ against the GDE with model (5.2) for data obtained on least squares meshes and the corresponding residuals vs. fitted plot.

	β_1	β_2	β_3	$\tilde{\mathtt{R}}^2$
nodal	1.3846	1.0106	1.2314	0.9975
H^1 modal	0.9106	1.0012	1.2287	0.9994
least squares modal	0.9175	1.0594	1.3802	0.9965

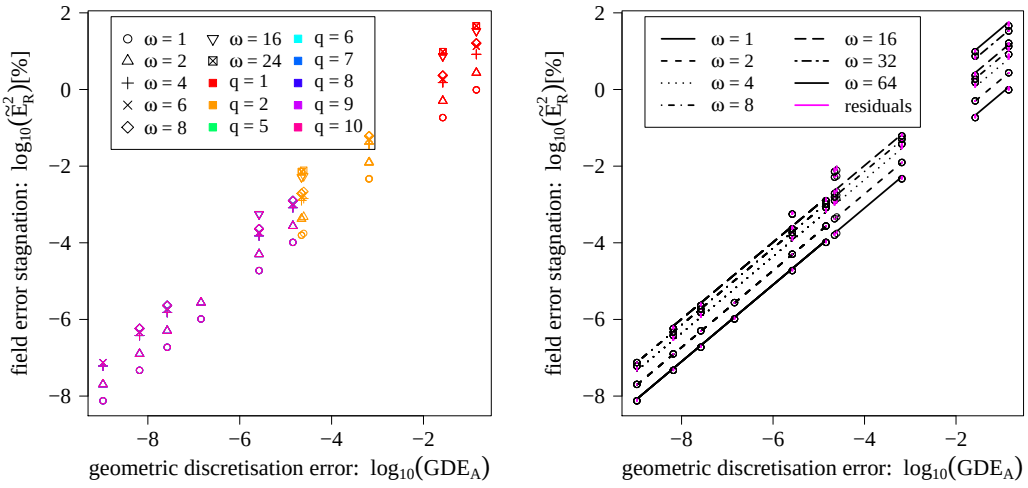
Table 5.3: Parameter values of the fits of model (5.2) to the field error evaluation and GDE definitions for simulations on H^1 modally curved meshes of orders $q \in \{1, 2, 5, 6, 7, 8, 9, 10\}$, for simulations on meshes curved with the modal least squares approach of orders $q \in \{1, 2, 3, \ldots, 10\}$, and the corresponding values of the adjusted $\mathtt{R}^2$. The parameter values obtained for the nodally curved meshes with $q \in \{1, 2, 3, 4\}$ are provided as reference.

corresponding plots are displayed with the modal least squares. For the H^1 modal curving, some data points catch the viewer's eye, in both, the data plots and the residuals vs. fitted plots. These are the data points for order $q = 3$ and some for $q = 4$ and will be considered separately. The remaining data for order $q \in \{1, 2, 5, 6, 7, 8, 9, 10\}$ show the linear behaviour as expected. It is separately plotted in Figure 5.18, along with a corresponding model fit and the residual vs. fitted plot. The latter shows that without the data for $q = 3, 4$, the considered model is suitable. The model parameters β_i of the fit as well as the adjusted $\mathtt{R}^2$ are presented in Table 5.3. The evaluated influences of the GDE and ω is comparable to those found for the nodally curved meshes. This is explained by the fact that the two parameters β_2 and β_3 are similar in both cases. But the offset parameter β_1 is significantly smaller for the results on the H^1 modally curved meshes. Neglecting the effects for $q = 3$ and $q = 4$, this means that for the same area-based GDE, less simulation error is obtained on H^1 modally curved meshes. As this is similar for the modal least squares results, the effect can be related to the modal shape functions.

The effects for $q = 3$ and $q = 4$ are explained in the following. In Section 4.4.3, it was observed that for the same simulation and mesh parameters, the H^1 modally curved meshes can lead to better field errors than the nodally curved meshes. In Figure 5.19, the data for $q = 3$ and $q = 4$ is shown separately. To improve the clarity of the plot, only selected ω values are depicted. The colour encodes the MpC-refinement, hollow symbols correspond to the simulations with $n_{\mathcal{C}}^e = 12$, solid symbols correspond to those with $n_{\mathcal{C}}^e = 28$.

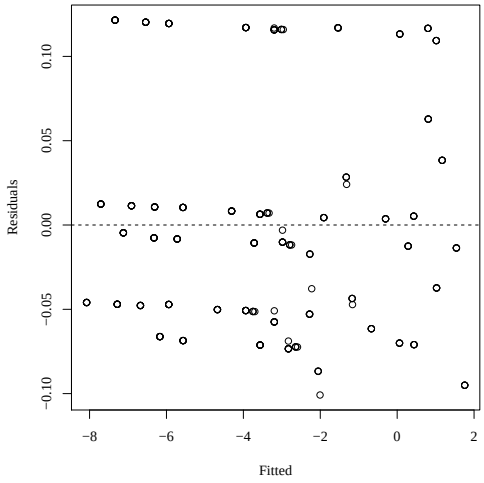
It can be observed that for $q = 3$, $n_{\mathcal{C}}^e = 28$, and varying MpC, the GDE_A is nearly constant, while the field error drops by more than 2 orders of magnitude from $\mathrm{MpC} = 100$ to $\mathrm{MpC} = 5000$. This indicates that on one side the Mmesh is already accurate enough with $\mathrm{MpC} = 100$ to allow the expected GDE_A. On the other side, not all the information that would be needed to describe the GIE-GDE relationship is included in this GDE measure. As the Mmesh is refined, the H^1 modal curving further improves some geometric aspect that has a positive effect on the GIE.

In acoustics simulations, the scattering direction is determined by the normal direction of the geometry. Further, the H^1 modal curving is based on the optimisation of the H^1-seminorm which corresponds to the L^2-norm of the derivatives. The behaviour is not observed with simulations on meshes that were curved with the least squares approach. All together, this suggests that a GDE measure based on the difference of the normal direction of the exact and the approximate geometry could improve the description of the GIE. Note that all corresponding measures we found in the literature were parametrisation dependent and not strictly proper, meaning they could be zero for non-matching curves (see Section 2.2.4). The development and analysis of a differential geometry based GDE measures without these disadvantages is left for



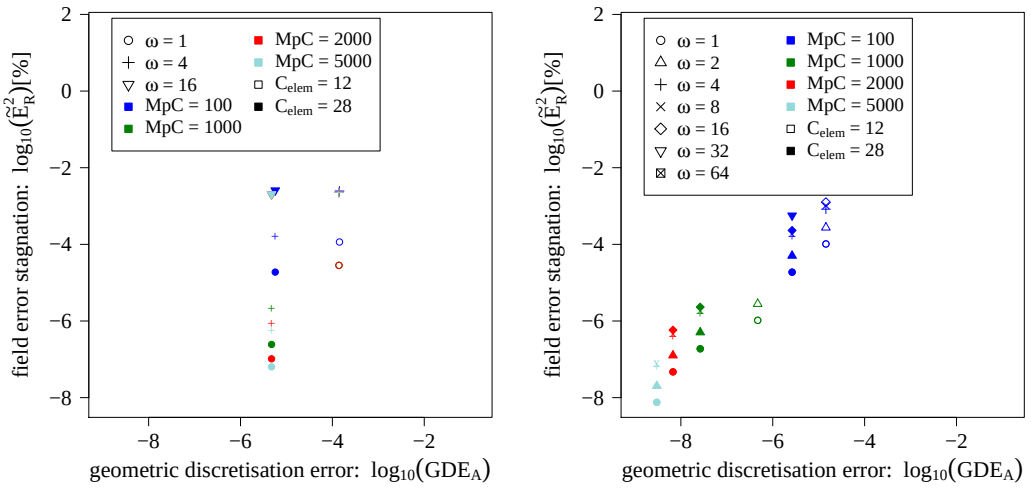
(a) data, without $q = 3, 4$

(b) fitted, without $q = 3, 4$



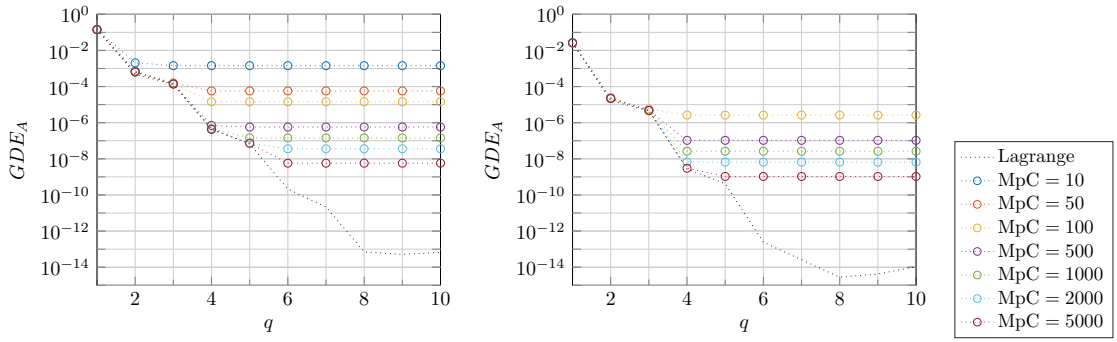
(c) fitted, without $q = 3, 4$

Figure 5.18: Plots of the fitting of the GIE $\tilde{E}_R^2$ against the GDE for H^1 modal curving, without data for $q = 3, 4$ with model (5.2). In (a), the colour decodes the geometric order of the mesh, while the shapes decode the ω value of the simulations.



(a) $q = 3$ (b) $q = 4$

Figure 5.19: Data for $q = 3$ and $q = 4$ of the GIE and GDE of the H^1 modally curved meshes. The colour decodes the MpC-refinement, the shapes decode the frequency ω. Filled symbols correspond to $n_{\mathcal{C}}^e = 12$, hollow symbols correspond to $n_{\mathcal{C}}^e = 28$.



(a) $n_{\mathcal{C}}^e = 12$ (b) $n_{\mathcal{C}}^e = 28$

Figure 5.20: q-convergence curves of GDE for the circle test case curved with the H^1 modal curving with (a) $n_{\mathcal{C}}^e = 12$ and (b) $n_{\mathcal{C}}^e = 28$.

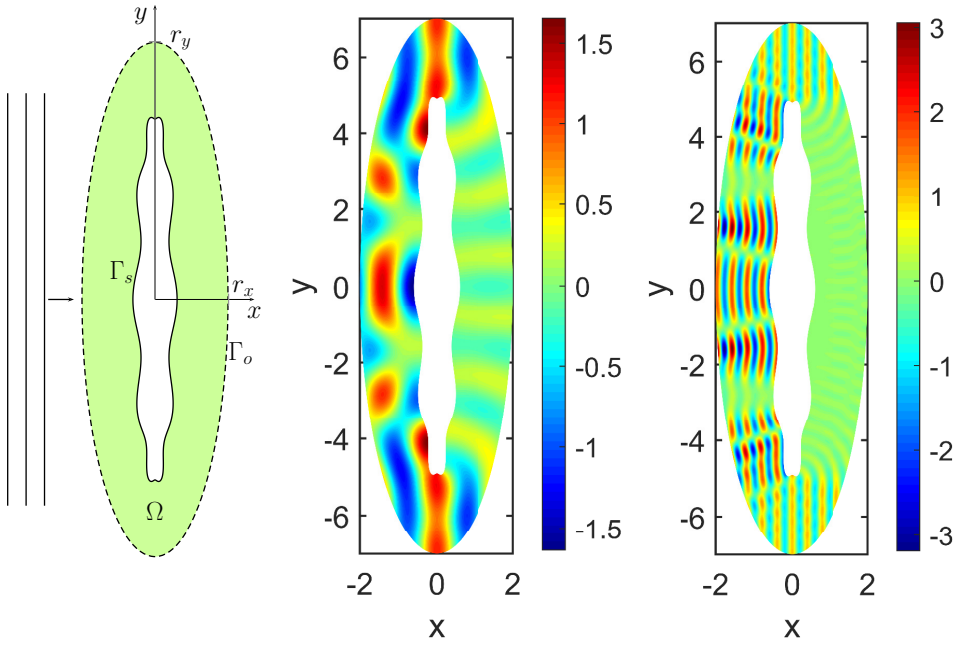
future work (see Section 8.2).

For $q = 4$, the described effect occurs only for high enough MpC-refinements, namely $\mathrm{MpC} = 1000$ for $n_{\mathcal{C}}^e = 12$ and $\mathrm{MpC} = 5000$ for $n_{\mathcal{C}}^e = 28$ (see Figure 5.19(b)). For lower refinements, the inaccuracy of the Mmesh is dominating the GDE_A measure (see Figure 5.20). The same applies for higher curving orders $q = 5 - 10$. This seems to indicate that the area-based GDE_A is still the dominant factor. As long as the MpC-refinement leads to an improvement of the GDE_A, the mentioned, so far unmeasured GDE component has no effect on the GDE-GIE relationship. It is only contributing once the optimal GDE_A is achieved.

5.5.2 Scattering of a Plane Wave by the Distorted Ellipse, H^1 Modal Curving

In this section, the model for the GDE-GIE relationship is tested for the scattering of a plane wave by a more complex geometry, namely the distorted ellipse that was also considered in Section 3.3.2.

Figure 5.21(a) shows a sketch of the domain. For the simulations, the domain parameters are



(a) (b) $\omega = 4$ (c) $\omega = 16$

Figure 5.21: (a) Sketch of the domain for the scattering test case and reference solutions for (b) $\omega = 4$ and (c) $\omega = 16$.

$r_x = 2$ and $r_y = 7$. As in the previous test case, the incident plane wave with wavenumber k is travelling in positive x-direction and is described by $\mathrm{p}_i = \mathrm{e}^{-ikx}$. As no analytical solution is available for this geometry, the characteristic boundary conditions along the outer boundary Γ_o are replaced by a perfectly matched layer (PML) (see [11], [10]). The reference solutions are obtained with adaptive field order on a highly refined quadratic mesh and shown in Figures 5.21(b) and 5.21(c). Analogously to the field error evaluation in the previous test case, the field error is evaluated as the relative L^2-error along an ellipse with $r_x = 1.25$ and $r_y = 6$.

	β_1	β_2	β_3	$\tilde{\mathrm{R}}^2$
H^1 modal	0.85762	1.0916	1.3861	0.9783

Table 5.4: Parameter values obtained for the fitted GDE-GIE model (5.2) to the data of the scattering by the distorted ellipse, and the corresponding value of the adjusted R^2.

In Figure 5.22, the data obtained from simulations, the model fit and the corresponding residuals vs. fitted plot are shown. The fitting parameters are summarised in Table 5.4. The fit for this test case is markedly worse than those for the circle test cases. However, the general model trend is still present, including the roughly linear dependency on the GDE and the sup-linear dependency on ω. From $q = 3$ to $q = 4$, the data drops, analogously to the drop observed for the circle test case with modally curved meshes. Therefore, it could correspond to the previously described limitation of the area based GDE. A lack of information in the model set-up is also suggested by

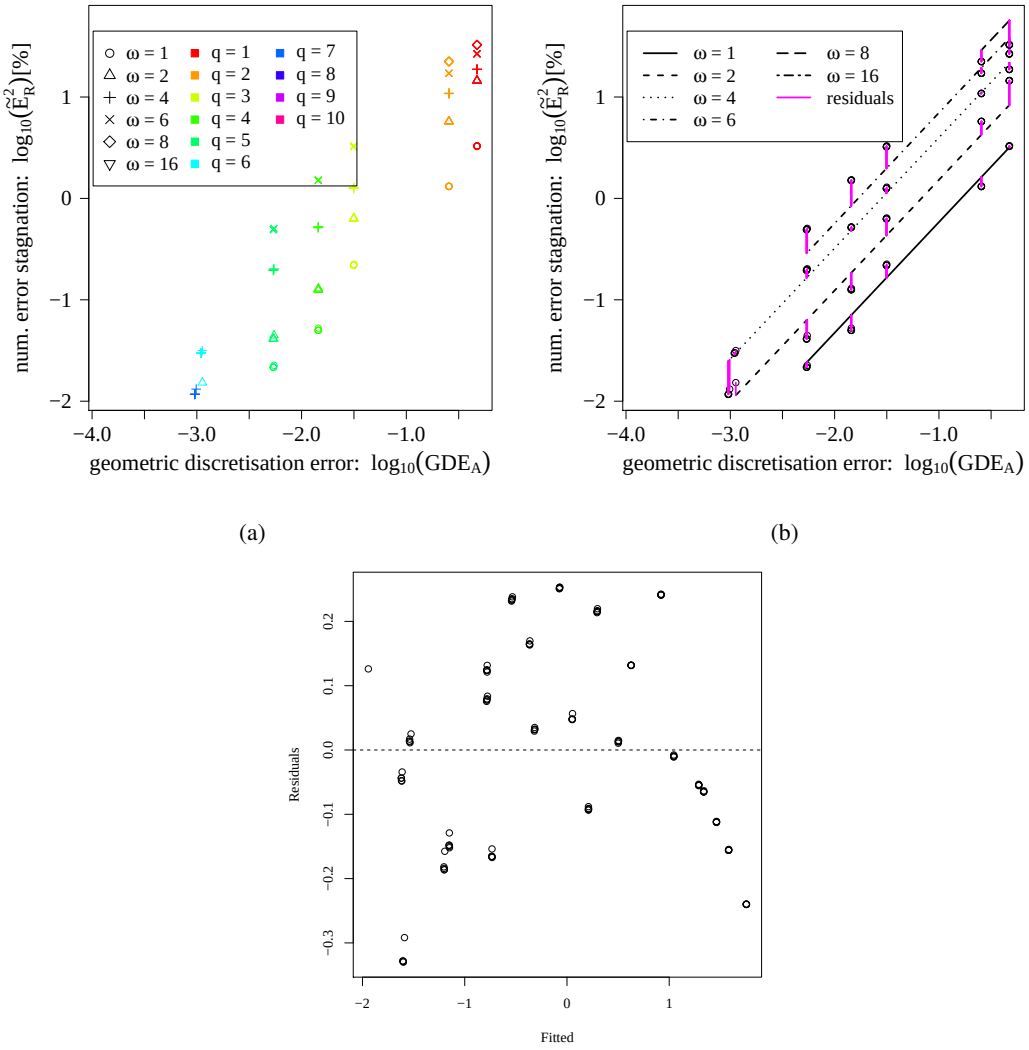


Figure 5.22: Plot of (a) the data set of the GIE $\tilde{E}_R^2$ against the GDE for the H^1 modal curving of the distorted ellipse test case. The different colours represent the geometric order of the mesh, while the different shapes represent the ω value of the simulations. In (b), the model fit and residuals are shown and in (c) the corresponding residuals vs. fitted plot.

the residuals vs. fitted plot, where a slight quadratic pattern in the residuals can be detected.

5.6 Conclusions of this Chapter

In this chapter, the relationship between the geometry discretisation error and the geometry induced error is evaluated for an acoustic scattering problem. In the literature, no such analysis was found. First, the nodally curved meshes are considered. Three GDE and three field error measures are tested for their effect on the evaluation. For those combinations with a clear link, a linear dependency of the GIE is found with the GDE and a super-linear dependency with the frequency ω. From the suggested measures, the mapping based GDE and the corresponding field error are dropped because they are impractical for general geometries. The projection based field error evaluation is rejected, as it does include an extra error in the calculation and does not

provide any additional information. For the tested geometry, the area based GDE measure showed the same behaviour as the Hausdorff distance. The latter is computationally more costly and is therefore dropped as well.
In the second part of the Chapter, the obtained relationship model is tested on H^1 modally curved meshes and meshes curved with the least squares approach. In general, the model provides a good fit for the modally curved meshes and the parameters imply that the same GDE leads to smaller GIE values than on nodally curved meshes. Further, it is observed that the considered area based GDE measure does not cover all aspects. As already seen in Chapter 4, the GIE drops on H^1 modally curved meshes in some cases without any indication in the current GDE measure. A similar sudden decrease in the GIE is observed for scattering simulations on the more complex geometry of the distorted ellipse. It is assumed that a combination of the area based GDE measure and a measure based on the normal direction could explain and model this behaviour. Overall, the results of Chapter 3-5 lead to the decision that the H^1 modal approach is the most advantageous of the tested curving approaches. It will be extended to 3D in the next chapter.

Chapter 6

Extension of the H^1 Modal Curving Algorithm to 3D

In this chapter, the extension of the modal curving algorithm described in Section 3.2.2.2 to 3D is outlined. After the presentation of the theoretical background, the individual curving steps are explained. To allow the curving of realistic geometries, the treatment of feature lines is added to the curving method. A sphere test case and a cylinder test case are considered to validate the implementations. Finally, the blending approach is adapted for the 3D meshes and tested with the scattering of a plane wave by a unit sphere.

6.1 Generalisation of Modal Curving

Compared to the 2D curving routine, the curving process in 3D becomes more evolved. In 2D the continuity between elements was enforced by default by the fixed vertices. For boundary faces in 3D, not only the continuity at the triangle vertices has to be ensured, but also along the triangle edges. While the former is again provided by the fixed vertex positions, the latter has to be taken care of in the setup of the element curving. As for nodally curved meshes, where the nodes along an edge are the same for both adjacent faces, here the α-coefficients must be the same (neglecting orientation based sign changes).

This would not be the case if the faces are curved one by one, considering the full triangular target surface. That is because edge shape functions have non-zero contribution on the element face. If the full triangular target surface is taken into account for the interpolation with the edge shape functions, the shape of the face element interior would influence the calculation of the α-coefficient. Further, this calculation would be independent of the shape of the neighbouring elements. Similarly, in the interpolation on the other edge adjacent face, the α-coefficients for the shape functions of the shared edge would only depend on the shape of the target triangle of this other face. The resulting curved faces would in general not have the same α-coefficient on the shared edge and the resulting surface would not be continuous along this edge.

In this section, the theory of the 3D curving is described and how the continuity issue is bypassed. Further, the implementation of the curving steps is outlined.

6.1.1 Theory of the 3D Curving

The calculation of the α values for the triangular faces on the domain boundary is split up in three steps: First the coefficients for the vertex shape functions are calculated, then those for the triangular edge shape functions, and finally the coefficients corresponding to the triangular bubble shape functions. The interpolant u_h of u on a triangular face is the sum of the vertex, edge, and bubble interpolants (see [82])

$$u_h = u_h^v + u_h^e + u_h^b.$$

Vertex Shape Functions

The vertex interpolant u_h^v is directly defined by the triangle vertices v_j and interpolates them exactly:

$$u_h^v(v_j) = u(v_j), \;\; j \in \{1, 2, 3\}\,.$$

Edge Shape Functions

The three edges are treated separately and the edge interpolant is defined as the sum of the individual edge interpolants

$$u_h^e = \sum_{j=1}^{3} u_h^{e_j}.$$

In the following, the interpolation of a single edge e_j is considered.

Based on the theory, the $H_{00}^{\frac{1}{2}}(e_j)$-norm should be considered for the interpolation of e_j ([82]). It is based on the trace of H^1-functions on the triangular element T with zero contribution on edges e_i, $i \neq j$, onto the boundary of the element, restricted to the edge e_j.

$$H_{00}^{\frac{1}{2}}(e_j) = \left\{ \tilde{w}|_{e_j}; \tilde{w} \in H^{\frac{1}{2}}(\partial T), \tilde{w} \equiv 0 \text{ on } \partial T \setminus e_j \right\},$$

where $H^{\frac{1}{2}}(\partial T)$ is the space of traces on ∂T from H^1-functions on T. For $\tilde{w} \in H_{00}^{\frac{1}{2}}(e_j)$, the norm is defined as

$$\|\tilde{w}\|_{H_{00}^{\frac{1}{2}}(e_j)} = \|w\|_{H^1(T)},$$

with $w|_{e_j} \equiv \tilde{w}$, $w|_{e_i} \equiv 0$ for $i \neq j$, and $\Delta w|_T = 0$. In other words, w is the expansion of $\tilde{w}$ onto the whole triangle, whose trace on edge e_j corresponds to $\tilde{w}$. The trace of w along the other two edges is 0, and w additionally fulfils the property $\Delta w = 0$ on the whole triangle.

The coefficients to solve the minimisation

$$\min \|\tilde{u} - \tilde{u}_h^v - \tilde{u}_h^{e_j}\|_{H_{00}^{\frac{1}{2}}(e_j)}$$

are determined by solving the system

$$(\tilde{u} - \tilde{u}_h^v - \tilde{u}_h^{e_j}, \tilde{\varphi}_{T,k}^{e_j})_{H_{00}^{\frac{1}{2}}(e_j)} = 0, \;\; k = 2, 3, \dots q^{e_j},$$

with $\tilde{u}_h^{e_j} = \sum_{m=2}^{q^{e_j}} \alpha_m^{e_j} \tilde{\varphi}_{T,m}^{e_j}$, and $(\cdot,\cdot)_{H_{00}^{\frac{1}{2}}(e_j)}$ the $H_{00}^{\frac{1}{2}}(e_j)$ scalar product.
For a simplification of the evaluation, a weighted $H_0^1(e_j)$-norm is suggested ([82]), where the weights correspond to the inverse of the Jacobian of the mapping from there reference line element to e_j. The difference between the two norms was found to be insignificant for $q < 10$ ($p < 10$) (see [19]). Due to the chosen weight, the weighted $H_0^1(e_j)$-norm simplifies to the standard $H_0^1(\hat{T}_{\text{line}})$-norm on the line reference element. For the considered shape functions $\tilde{\varphi}_{T,k}^{e_j}$ with $k = 2, 3, \dots q^{e_j}$, this corresponds to the $H^1(\hat{T}_{\text{line}})$-seminorm. It is the same that was already considered for the α calculations in the 2D case. Therefore, analogously to the 2D derivation (see Section 3.2.2.2), the formula to find the edge α_k value is

$$\begin{aligned}
& (\tilde{u} - \tilde{u}_h^v - \tilde{u}_h^{e_j}, \tilde{\varphi}_{T,k}^{e_j})_{H_0^1(\hat{T}_{\text{line}})} = 0 \\
\Leftrightarrow \; & (\tilde{u}, \tilde{\varphi}_{T,k}^{e_j})_{H_0^1(\hat{T}_{\text{line}})} - (\tilde{u}_h^v, \tilde{\varphi}_{T,k}^{e_j})_{H_0^1(\hat{T}_{\text{line}})} - (\tilde{u}_h^{e_j}, \tilde{\varphi}_{T,k}^{e_j})_{H_0^1(\hat{T}_{\text{line}})} = 0 \\
\Leftrightarrow \; & (\tilde{u}, \tilde{\varphi}_{T,k}^{e_j})_{H_0^1(\hat{T}_{\text{line}})} = \alpha_k (\tilde{\varphi}_{T,k}^{e_j}, \tilde{\varphi}_{T,k}^{e_j})_{H_0^1(\hat{T}_{\text{line}})} \\
\Leftrightarrow \; & \alpha_k = (\tilde{u}, l_k)_{H_0^1(\hat{T}_{\text{line}})} \\
\Leftrightarrow \; & \alpha_k^{e_j} = \int_{-1}^{1} \tilde{u}' l_k'.
\end{aligned}$$

Note that by curving each edge independently of the element face deflection, the continuity between elements is ensured.

Bubble Shape Functions

The coefficients α_{n_1,n_2}^b for the face bubble interpolant $u_h^b = \sum_{n_1=1}^{q^b-2} \sum_{n_2=1}^{q^b-n_1-1} \alpha_{n_1,n_2}^b \varphi_{T,n_1,n_2}^b$ are chosen to minimise the residual of the overall interpolation in the $H^1(\hat{T}_{tria})$ seminorm ([82]):

$$\begin{aligned}
& \min |u - u_h^v - u_h^e - u_h^b|_{H^1(\hat{T}_{tria})} = \\
& \quad \min \int_{\hat{T}_{tria}} \nabla(u - u_h^v - u_h^e - u_h^b) \cdot \nabla(u - u_h^v - u_h^e - u_h^b) \, \mathrm{d}\boldsymbol{\xi}
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \frac{\partial}{\partial \alpha_{n_1,n_2}^b} & \int_{\hat{T}_{tria}} \nabla(u - u_h^v - u_h^e - u_h^b) \cdot \nabla(u - u_h^v - u_h^e - u_h^b) \, \mathrm{d}\boldsymbol{\xi} \\
& = 2 \int_{\hat{T}_{tria}} \nabla(u - u_h^v - u_h^e - u_h^b) \frac{\partial}{\partial \alpha} \nabla(u - u_h^v - u_h^e - u_h^b) \, \mathrm{d}\boldsymbol{\xi} \\
& = 2 \int_{\hat{T}_{tria}} \nabla(u - u_h^v - u_h^e - u_h^b) \nabla \varphi_{T,n_1,n_2}^b \, \mathrm{d}\boldsymbol{\xi} \\
& = 0
\end{aligned}$$

$$\Rightarrow \int_{\hat{T}_{tria}} \nabla(u - u_h^v - u_h^e)\nabla\varphi^b_{T,n_1,n_2}\,\mathrm{d}\boldsymbol{\xi} = \int_{\hat{T}_{tria}} \nabla u_h^b \nabla\varphi^b_{T,n_1,n_2}\,\mathrm{d}\boldsymbol{\xi},$$

with φ^b_{T,n_1n_2} the bubble shape functions. This results in the system

$$\begin{aligned}
&\boldsymbol{A\alpha} = \boldsymbol{b}, \text{ with}\\
&A_{i,j} = \int_{\hat{T}_{tria}} \nabla\varphi^b_{T,n_{1,i},n_{2,i}} \nabla\varphi^b_{T,n_{1,j},n_{2,j}}\,\mathrm{d}\boldsymbol{\xi},\\
&b_i = \int_{\hat{T}_{tria}} \nabla(u - u_h^v - u_h^e)\nabla\varphi^b_{T,n_{1,i},n_{2,i}}\,\mathrm{d}\boldsymbol{\xi}.
\end{aligned}$$

Note that this is not an explicit formula for the $\alpha^b_{n_1n_2}$, as the bubble shape functions do not fulfill the δ-property. In [83], bubble shape functions are constructed that fulfill the δ-property. As all bubble shape functions vanish at the boundary of the reference triangle, they can be orthonormalised in the $H^1_0(\hat{T}_{tria})$-norm, using a modified Gram-Schmidt algorithm. With these modified bubble shape functions $\tilde{\varphi}^b_{T,n_1,n_2}$, the calculation of the corresponding $\tilde{\alpha}^b_{n_1n_2}$ simplifies to

$$\tilde{\alpha}^b_{n_1,n_2} = \int_{\hat{T}_{tria}} \nabla(u - u_h^v - u_h^e)\nabla\tilde{\varphi}^b_{T,n_1,n_2}\,\mathrm{d}\boldsymbol{\xi}.$$

In this thesis, the originally introduced, non-orthonormalised bubble shape functions will be considered.

6.1.2 Implementation of the Edge Curving

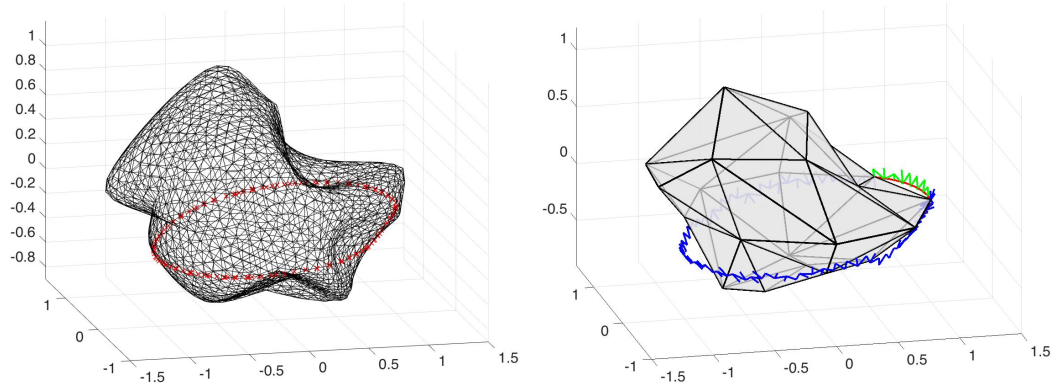
For the edge curving, each edge is treated separately. It is curved using the 2D line shape functions and the algorithm described for the 2D boundary line elements (see Section 3.2.2.2). In the 2D case, the target curve for the edge was defined by the linear interpolation of the vertices of the Mmesh arc over the Cmesh element. In 3D, it is still ensured that the Cmesh vertices coincide with Mmesh vertices. But it is generally not the case that projections of Mmesh edges are subsets of a Cmesh edge. Instead, the target curve for each edge has to be detected as a preprocessing step.

6.1.2.1 Edge Target Curve Detection

The edge target curve detection consists of two steps. In the first step, possible target points have to be found that lie on the Mmesh, the general target surface. As a unique definition, all intersections of the Mmesh with the edge normal plane

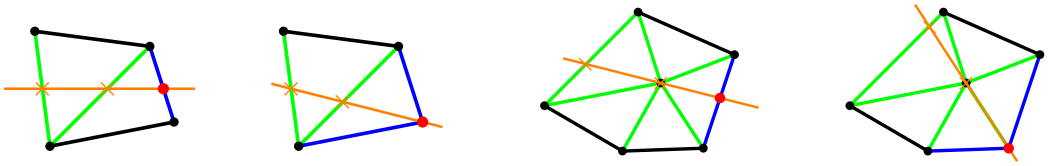
$$\boldsymbol{x} = \boldsymbol{p_1} + r(\boldsymbol{p_2} - \boldsymbol{p_1}) + s\boldsymbol{n_e},$$

are chosen, with $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ the two vertices of the Cmesh edge, $\boldsymbol{n_e}$ the edge normal, and $r, s \in \mathbb{R}$. The edge normal is defined as the mean of the normals of the two adjacent faces. The normal plane is chosen as it intuitively matches the idea of dividing the space in the middle between these two faces. An example of a set of intersection points for a single Cmesh edge and the corresponding Mmesh edges is shown in Figure 6.1.



(a) intersection points (b) intersecting edges

Figure 6.1: Mmesh with the intersection points (red crosses) considered for the Cmesh edge target curve, and the Cmesh with the corresponding Mmesh edges (blue lines) and the approximation after edge curving (red line).



(a) (b) (c) (d)

Figure 6.2: The four different possible cases for the edge traversal. Already found intersection points are marked by a orange cross. The orange line indicates the intersecting plane. Green lines show Mmesh edges that were already considered in previous steps, black lines are Mmesh edges without intersections, and blue lines are the remaining Mmesh edges, therefore those on which the next intersection point lies. This next intersection point is indicated as red dot.

This way, far too many intersection points are found. In the second step of the edge target curve detection, the actual target points are selected with a traversal algorithm. For this algorithm, not only the intersection points but also the corresponding Mmesh edges are considered. Starting from one Cmesh edge vertex, a path of connected, preselected Mmesh edges is constructed.

In general, there are four possible situations that have to be considered when looking for the next intersection point or edge (see Figure 6.2): The currently considered point can either coincide with an Mmesh vertex or lie on the interior of an Mmesh edge. Similarly, the follow-up intersection point can fall together with an Mmesh vertex, or simply on an Mmesh edge. The treatment of the different cases is detailed in Appendix B.

The Mmesh edge traversal is done for all possible starting edges, until the paths either run into a deadend or until they reach the second Cmesh edge vertex. In the end, the shortest full path is selected. The corresponding intersection points are extracted and their linear interpolation forms the edge target curve.

6.1.3 Implementation of the Face Curving

Similar as the edge curving, the face curving follows several steps. The first step is the detection of the face target points. Before a parametrisation of the target face can be constructed, a triangulation of the target points has to be found. It is based on the underlying Mmesh structure. This triangulation is then also used to ensure a piecewise exact integration over the Cmesh element in the calculation of the $\boldsymbol{\alpha}$-values.

6.1.3.1 Detection of Face Target Points

As face target points, all Mmesh vertices have to be detected that lie within the boundary formed by the Cmesh edge normal planes. This is done with a node traversal algorithm, following the node neighbour relationship. The status of each Mmesh vertex is described by two properties s_1 and s_2, where

$$s_1 = \begin{cases} 0 & \widehat{=} \text{ vertex not tested (default)} \\ 1 & \widehat{=} \text{ vertex lies inside} \\ 6 & \widehat{=} \text{ vertex lies on plane defining edge target curve} \\ -1 & \widehat{=} \text{ vertex lies outside} \end{cases} \quad \text{and} \quad s_2 = \begin{cases} 0 \ \widehat{=} \text{ neighbouring nodes not checked (default)} \\ 1 \ \widehat{=} \text{ neighbouring nodes checked} \end{cases} .$$

The seemingly arbitrary choice of the number 6 to decode vertices on the curve is justified by the subsequent decoding of resulting triangles (see Sec. 6.1.3.2).
First, the three Cmesh edge normal planes are considered. The signed distance between the three planes and the vertices of the intersecting Mmesh edges are evaluated. This way, these vertices can be classified as face internal vertices ($s_1 = 1$), face external vertices ($s_1 = -1$), or vertices that fall onto one of the planes ($s_1 = 6$). The latter is especially the case for those Mmesh vertices that collide with the Cmesh vertices. Note that the orientations of the Cmesh edges has to be included in the calculation. Further, some vertices might be considered as ”inside” from one of the edge normal planes, but ”outside” from another. For this reason, each of the Mmesh vertices in question has to be tested, even if it was already marked ”inside” with respect to a previously tested plane. If a Mmesh vertex is detected as ”outside” with respect to one of the Cmesh edges, it is defined as ”outside” for the corresponding Cmesh face.
Starting from this bounding data set, the nodes with status $s_1 = 1$ and $s_2 = 0$ are traversed. Their neighbours in the Mmesh data structure are extracted and, if they were not tested yet ($s_1 = 0$), are set to be an interior node. Once no more node exists with $s_1 = 1$ and $s_2 = 0$, the selection algorithm terminates. All Mmesh vertices with $s_1 = 1$ or $s_1 = 6$ are selected as face internal target points.

6.1.3.2 Triangulation

In order to construct a parametrisation of the face target as well as a piecewise exact integration, a triangulation of the internal target points is required. Therefore, all Mmesh faces are preselected that have at least one vertex among the internal target points. They are classified by the sum $S_1 = \sum_{i=1}^{3} s_1(\boldsymbol{v_i})$ of the s_1 status of their vertices $\boldsymbol{v_i}$:

- All-in faces ($S_1 = 3$)
- 2-in-1-boundary faces ($S_1 = 8$)
- 1-in-2-boundary faces ($S_1 = 13$)
- All-boundary faces ($S_1 = 18$)
- 2-in-1-out faces ($S_1 = 1$)
- 1-in-1-boundary-1-out faces ($S_1 = 6$)
- 2-boundary-1-out faces ($S_1 = 11$)
- 1-in-2-out faces ($S_1 = -1$)
- 1-boundary-2-out faces ($S_1 = 4$)

Additionally to these cases, all Mmesh triangles that have no interior node but at least one intersection with the Cmesh edge normal planes are considered.
Each of the classes is treated separately. For All-in, 2-in-1-boundary, 1-in-2-boundary, and All-boundary faces, the Mmesh structure can be preserved. All the other classes are further split in different cases, depending on the intersection with the edge normal planes. Non-triangular faces are retriangulated. The detailed cases and treatment of the different classes is provided in Appendix C.
In the end, small triangles are merged with neighbouring triangles. First, nodes with distance less than 10^{-11} are merged and all triangles with the same vertex twice are deleted. Afterwards, all triangles that are detected as collinear are collapsed and deleted as well. The collinearity is evaluated using the algorithm described in [64] and a tolerance of 10^{-14}. During the entire process, vertices on the domain boundary need to be preserved.

6.1.3.3 Construction of the Parametrisation

To find a parametrisation of the triangulation, mean value coordinates as described in [27, 28] are used. They are length ratio and angle ratio preserving and correspond to the idea of length-based parametrisation in 2D.
First, the reference nodes for the vertices along on the Cmesh edges are found with a length-based parametrisation of the reference element edges. In the second step, the reference nodes of the remaining vertices are calculated by solving the linear systems

$$\begin{aligned} \boldsymbol{A}\xi &= \boldsymbol{b_\xi} \\ \boldsymbol{A}\eta &= \boldsymbol{b_\eta}, \end{aligned}$$

where

$$\begin{aligned} a_{i,j} &= \begin{cases} 1 & \text{if } i = j \\ -\lambda_{i,j} & \text{if vertex } \boldsymbol{v_j} \text{ is a neighbour of vertex } \boldsymbol{v_i} \\ 0 & \text{else,} \end{cases} \\ b_{\xi,i} &= \sum_{\boldsymbol{v_j} \in N_{i,B}} \lambda_{i,j} \xi_{v_j} \\ b_{\eta,i} &= \sum_{\boldsymbol{v_j} \in N_{i,B}} \lambda_{i,j} \eta_{v_j} \end{aligned}$$

with weights $\lambda_{i,j}$ and $N_{i,B}$ the set of neighbouring vertices of vertex $\boldsymbol{v_i}$ that lie on a Cmesh edge (see [27]). Two vertices are defined as *neighbouring vertices* if they are connected by an edge of

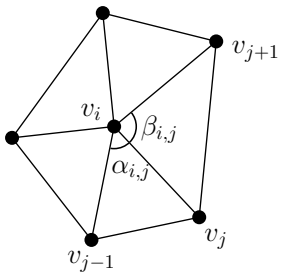


Figure 6.3: Sketch of the angles to find the weights for the mean value coordinates. This figure is based on Figure 1 in [28].

the triangulation. The weights $\lambda_{i,j}$ are calculated as

$$\lambda_{i,j} = \frac{\omega_{i,j}}{\sum_{\boldsymbol{v_k} \in N_i} \omega_{i,k}}$$
$$\omega_{i,j} = \frac{\tan(\frac{\alpha_{i,j}}{2}) + \tan(\frac{\beta_{i,j}}{2})}{||\boldsymbol{v_j} - \boldsymbol{v_i}||}$$

with $\alpha_{i,j}$ and $\beta_{i,j}$ the angles at $\boldsymbol{v_i}$ on both sides of the edge connecting $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ (see Fig. 6.3) and N_i the set of neighbouring vertices of vertex $\boldsymbol{v_i}$ (see [28]).
After the parametrisation, a test for colinearity is run for the triangles in the reference domain. Analogously to the check in the triangulation routine, colinear triangles are collapsed. The nodes and the triangles in the physical domain are treated correspondingly.

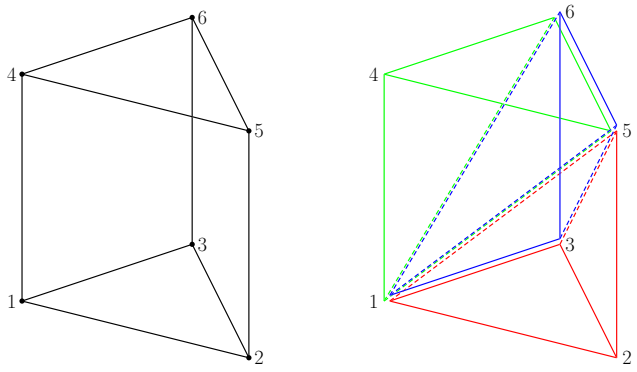
6.1.3.4 Piecewise, Exact Integration

The triangulation of the Mmesh arc is not only used to find a parametrisation, but also to construct an exact quadrature rule for the integration in the calculation of the $\boldsymbol{\alpha}$.
In general, the Mmesh arc is only piecewise linear. This makes the standard Gaussian quadrature inapplicable for integrations. Instead, an exact integration rule can be obtained by composing local Gaussian quadratures over each of the triangles of the arc triangulation. To obtain the quadrature points on these subtriangles, the Gauss-Legendre points on the $(-1, 1)$ reference triangle are mapped using a linear Lagrangian mapping. The quadrature weights are the corresponding Gauss-Legendre weights, scaled by $\frac{1}{2}A_e$, where A_e is the area of the subtriangle. Note that the additional reference mappings from the reference triangle to the subtriangles in the reference domain have to be also considered in the calculation of the integrand.

6.1.4 GDE Measure in 3D

Analogously to the area measure in 2D, the GDE measure in 3D is defined as the volume V_{approx} between the approximation and the target surface, divided by the surface area A_{target} of the target surface:

$$\text{GDE}_V = V_{\text{approx}} / A_{\text{target}}.$$



(a) wedge (b) subdivision

Figure 6.4: Sketch of the subdivision of a wedge into three tetrahedrons formed by the wedge nodes 1-2-3-5, 1-4-5-6, and 1-3-5-6.

In general, the Mmesh is considered as the target curve. But, as in 2D, some special cases allow the evaluation of the GDE_V with respect to the exact geometry.

Special case: Sphere

The sphere in 3D corresponds to the circle case in 2D (see Sec. 3.3.1.2). The knowledge of the geometry and the inverse of its parametrisation allows for a calculation of the volume between the approximating curve and the exact sphere. The resulting measure will be denoted GDE_V^S.
In 2D, the area formula for convex quadrangles could be manipulated to obtain an ODE describing the total area measure. In 3D, no such simple formula was found.
Instead, the idea corresponds to the more general idea that was followed in 2D for the area between the approximation and the Mmesh. There, the area was subdivided in small quadrangles of which the areas could be obtained by the previously mentioned formula. The quadrangles are refined until convergence is observed (see Section 3.3.2.1).
For the 3D volume between the approximation and the sphere, the volume is subdivided in small wedges. This is done by triangulation of the approximation based on a regular triangulation of the reference triangle. Corresponding points on the sphere were found by radial projection. A general formula for the volume of a wedge was not found in standard formularies. Instead, the volume is obtained by subdividing the wedge in three tetrahedras (see Fig. 6.4) and adding the volumes of these. This way, a formula is obtained to directly calculate the volume of the wedge. Note that this approach is only possible if we know that the nodes of the quadrangular wedge faces are coplanar. For the volume to sphere case, this is ensured by the way the wedges are created.

General GDE measure

To generalise the volume to the Sphere to the volume to the Mmesh, the construction of the wedges has to be adapted. The idea is to now rely on the triangulation of the reference triangle from the curving step. In this triangulation, each (small) triangle corresponds to a linear surface

on the Mmesh. No Mmesh edges are crossing them. At the same time, a parametrisation of both the curved Cmesh and the Mmesh are available from the triangulated reference triangle.
For these general wedges, there is no guarantee that the vertices defining a quadrangle face are coplanar. The subdivision in tetrahedra would therefore demand bookkeeping to ensure that the faces are subdivided by the same diagonal for both adjacent wedges. Instead, the volume of a single wedge W is calculated using the Green's identity:

$$\begin{aligned} V_{\text{approx}} &= \int_W 1 \,\mathrm{d}W = \int_W \nabla \cdot \boldsymbol{f} \,\mathrm{d}W \\ &= \int_{\partial W} \boldsymbol{f} \cdot \boldsymbol{n} \,\mathrm{d}(\partial W), \end{aligned}$$

with

$$\boldsymbol{f} = \frac{1}{3} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

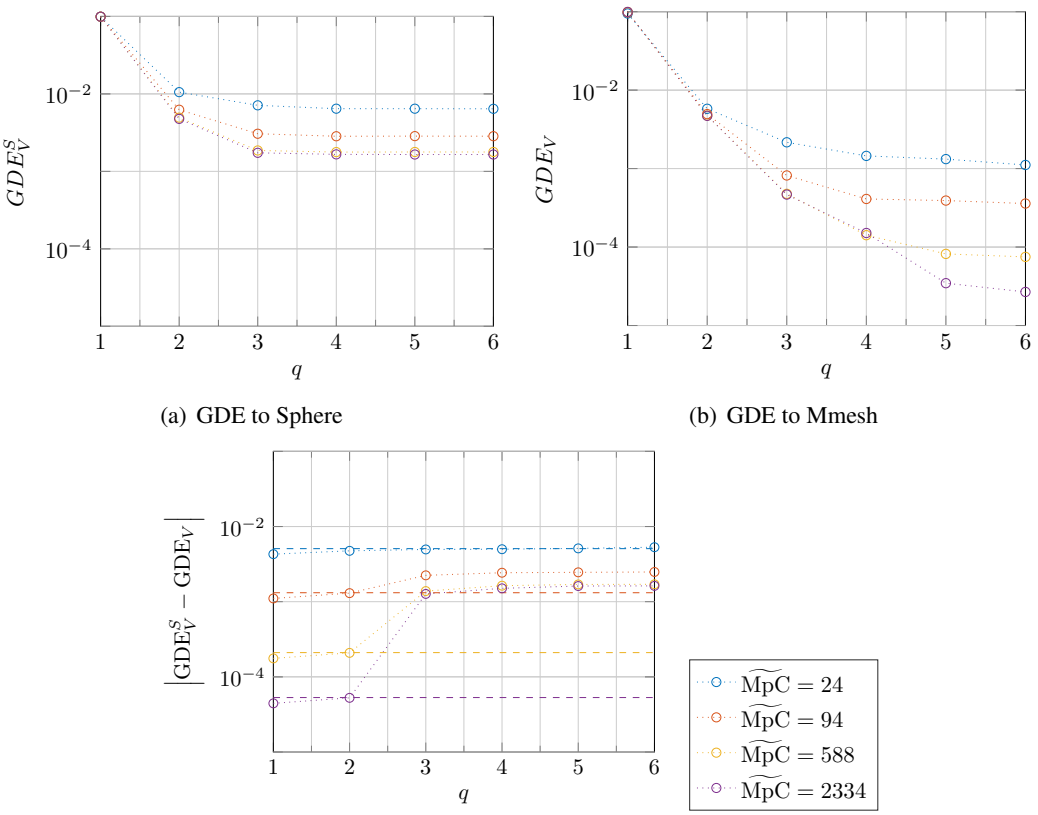
and $\boldsymbol{n}$ the outwards pointing unit normal of the wedge surface. Note that the norms of wedges that cross intersections of the Cmesh and Mmesh are not continuously pointing outwards, which makes Green's identity invalid for these wedges. As the subdivision into wedges is refined, the overall volume calculation is converging towards the correct volume.

6.1.5 Test Case: Curving of a Sphere

First, the 3D curving algorithm is tested for the unit sphere. Note that for 3D, the definition of the $\widetilde{\mathrm{MpC}}$-ratio is adapted. In 2D, the ratio was evaluated as the number of Mmesh edges per Cmesh edge and therefore directly linked to the element length. The analogous ratio in 3D is defined as the average number of Mmesh faces per Cmesh face

$$\widetilde{\mathrm{MpC}} = {}^{n^f_{\mathcal{M}}}/_{n^f_{\mathcal{C}}},$$

with $n^f_{\mathcal{M}}$ the number of $\mathcal{M}$ faces and $n^f_{\mathcal{C}}$ the number of $\mathcal{C}$ faces. The latter scale quadratically with the mesh length and could not be explicitly controlled by the mesh length in the mesh generation process. Therefore, the here considered $\widetilde{\mathrm{MpC}}$-ratios are not the same for the different $\mathcal{C}$ sizes, but were selected to lie relatively close together.
In Figures 6.5 and 6.6, the results of the GDE evaluation towards the exact sphere and towards the Mmesh are shown and compared for a coarse and a fine Cmesh. First of all, the 3D curving implementation is validated by the expected, typical error behaviour: An exponential convergence, ending in a stagnation, where the stagnation level decreases with increased geometric order q and increased number of elements $n^f_{\mathcal{C}}$.
Comparing the evaluations of the GDE measure towards the exact Sphere with the one towards the Mmesh, it can be observed that for the coarse Cmesh the two measures differ more than for the fine mesh. This is also consistent with the expectations, as the exact Sphere and the Mmesh lie closer together in the latter case (see dashed lines Figures 6.5(c) and 6.6(c)).

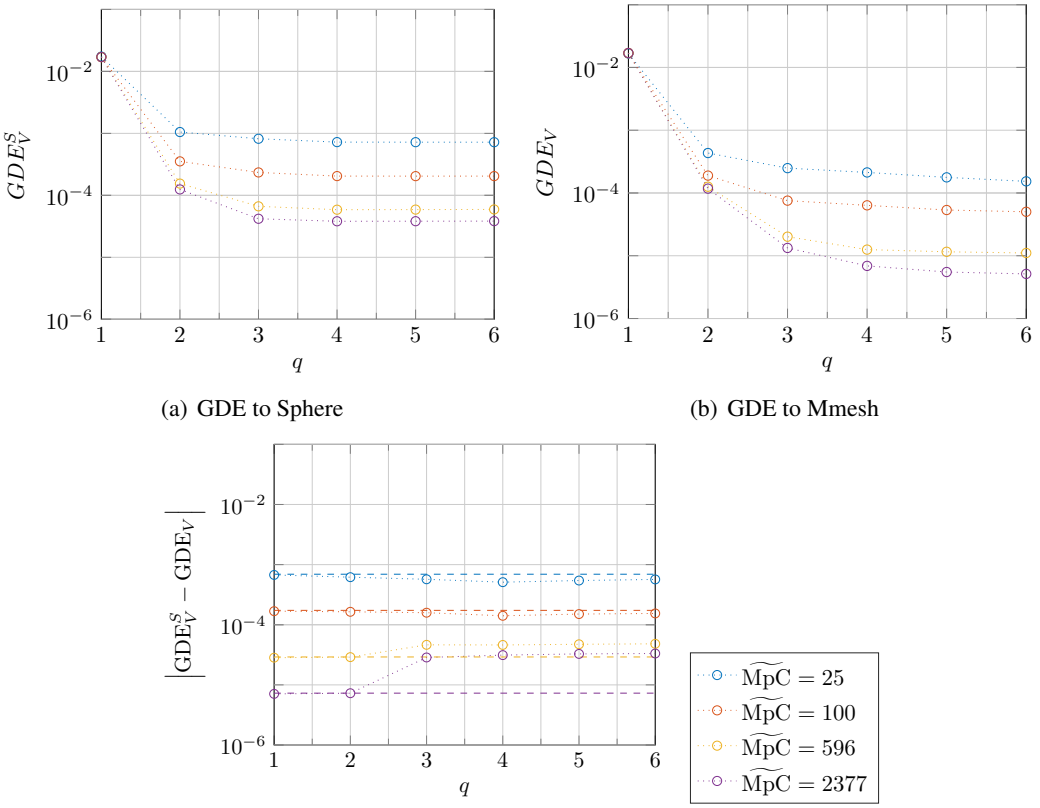


(a) GDE to Sphere

(b) GDE to Mmesh

(c) Difference in GDE evaluations

Figure 6.5: Comparison of the GDE evaluation to the exact Sphere and to the Mmesh for $n_{\mathcal{C}}^f = 32$. The dashed lines in (c) indicate the volume between the Mmesh and the exact Sphere.



(a) GDE to Sphere

(b) GDE to Mmesh

(c) Difference in GDE evaluations

Figure 6.6: Comparison of the GDE evaluation to the exact Sphere and to the Mmesh for $n_C^f = 228$. The dashed lines in (c) indicate the volume between the Mmesh and the exact Sphere.

But especially on the coarse Cmesh it becomes obvious that the GDE_V^S stagnates at a higher level than the GDE_V. An error component that is either not present or not significant in GDE_V becomes the dominating factor. In the context of this thesis, no explanation for this behaviour was found. It is observed though that the difference between the two measures deviates from the difference between the Mmesh and the exact sphere abruptly from $q = 3$ on. From this curving order on, the shape functions become non-convex and triangular bubble shape functions arise. Therefore, it is assumed that the unexplained error component is linked to at least one of these two properties. Considering Figure 6.5(c), the difference between the GDE_V^S and GDE_V is plotted and the volume between the exact Sphere and the Mmesh indicated by the dashed lines. Generally, one would expect that the difference between the two GDE measures corresponds to the volume between their two target curves. For $q = 1, 2$, this is the case, while for $q \geq 3$ the difference also remains approximately constant, but at a higher level. The jump between the two levels takes place between $q = 2$ and $q = 3$, just were the mentioned non-convex higher-order shape functions, as well as triangular bubble shape functions arise.
Note that results in Section 6.3.1 below will indicate that the GDE_V evaluated towards the Mmesh is the more reliable measure of the geometric accuracy regarding the corresponding geometry induced field error. For this reason, it will also be considered in the validation test case of the following section.

6.2 Feature Line Treatment

Realistic geometries do usually consist not only of one smooth surface, but are build of several surfaces that intersect each other. These intersections typically form sharp edges, which are called *feature lines*. For a reasonable geometric approximation, it is crucial to take these feature lines into account. In order to preserve feature lines, their occurrence on the geometry has to be known. Algorithms to detect feature lines based on e.g. the angle between neighbouring mesh elements are available in the literature ([47]).
For this thesis, the feature line detection is simplified. Here, only geometries are considered, where feature lines form closed loops around predefined surface patches. The knowledge which surface element belongs to which surface patch has therefore to be additionally provided along with the mesh. To check if an edge is on a feature line, the two adjacent surface elements are considered. If they belong to two different surface patches, the edge is marked as falling onto a feature line.
In the curving process, the feature lines have to be treated separately at only one point, namely the detection of the edge target points. The added knowledge of the feature line leads to a simplification of the detection step as it was described in Section 6.1.2.1: First of all, only Mmesh edges that are marked as falling onto a feature line are considered. This reduces the number of Mmesh edges that need to be tested. Secondly, the Cmesh and Mmesh are both expected to respect all feature lines. Therefore, one knows that all intersection points of the Mmesh and the Cmesh edge normal plane coincide with Mmesh vertices. And as feature lines can only cross in Cmesh vertices, only one possible next Mmesh edge can be found as long as travelling along the path into the right direction. If more than one possible next edge is detected, these do not have to be treated, but the path is either at the end vertex or can be rejected.

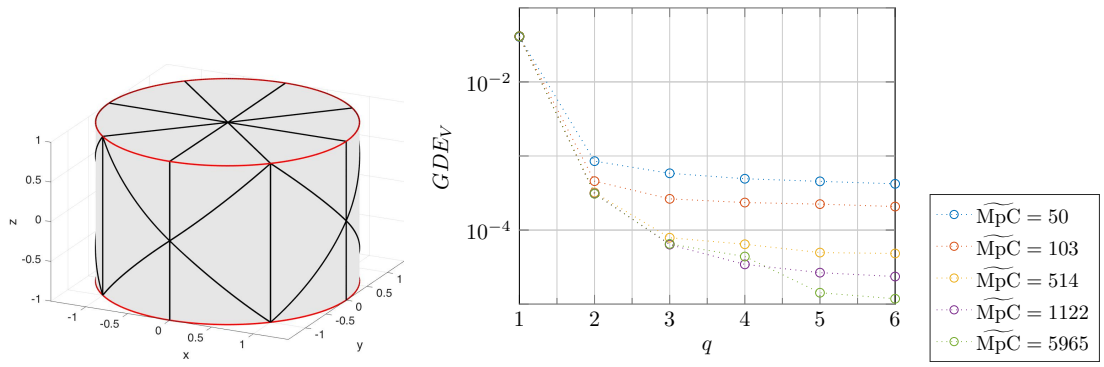


Figure 6.7: Geometry and GDE_V of the cylinder test case with $n_{\mathcal{C}}^f = 40$. The figure of the geometry shows the curved mesh with $\widetilde{\mathrm{MpC}} = 115$ and $q = 6$. The feature lines are highlighted in red.

Once the adjusted edge target curve detection is applied, the edge curving procedure remains the same as for Cmesh edges that fall not onto a feature line. The curving of the element faces is not affected by the feature line treatment.

6.2.1 Test Case: Cylinder

In order to validate the described feature line treatment, the curving of a cylinder is considered. The curving results are presented in Figure 6.7, along with a plot of the curved mesh. The feature lines are highlighted in red. As expected, the GDE_V first converges and then stagnates with increasing curving order q. The stagnation levels decrease with refinement of the target curve $\mathcal{M}$. Only for $\widetilde{\mathrm{MpC}} = 5965$, $q = 4$ an outlier is detected, where the GDE_V is higher than for the less refined target curve with $\widetilde{\mathrm{MpC}} = 1122$.

6.3 Blending Approach in 3D

The blending of the curved surface for the mapping of the full tetrahedral is analogous to the blending approach for the H^1 modally curved meshes in 2D (see Sec. 4.3.2). Starting from the linear interpolation of the tetrahedral vertices, the deviation of first the edges and then the faces is considered in each coordinate separately. Similar to the 2D triangle shape functions, the 3D tetrahedral Lobatto shape functions are constructed as vertex, edge, face and bubble shape functions, where the vertex shape functions coincide with the linear Lagrange shape functions. The traces of the edge and face shape functions along the corresponding edges and faces equal the 1D and 2D Lobatto shape functions and vanish on the remaining edges and faces, respectively (see [82]). Therefore, the tetrahedral Lobatto shape functions can be directly applied to the $\boldsymbol{\alpha}$-values obtained from the curving method. The mapping has therefore the form

$$\boldsymbol{f}(\xi,\eta) = \sum_{i=1}^{4} \boldsymbol{x}_i \varphi_{\mathrm{Tet}}^{v_i}(\xi,\eta) + \sum_{i=1}^{6}\sum_{j=2}^{q} \varphi_{\mathrm{Tet},j}^{e_i}(\xi,\eta)\,\boldsymbol{\alpha}_j^{e_i} + \sum_{i=1}^{4}\sum_{j=1}^{q-2}\sum_{k=1}^{q-j-1} \varphi_{\mathrm{Tet},j,k}^{f_i}(\xi,\eta)\,\boldsymbol{\alpha}_{j,k}^{f_i},$$

where $\varphi_{\mathrm{Tet}}^{v_i}$, $\varphi_{\mathrm{Tet},j}^{e_i}$, $\varphi_{\mathrm{Tet},j,k}^{f_i}$ are the tetrahedral vertex, edge, and face shape functions, and $\boldsymbol{\alpha}_j^{e_i}$, $\boldsymbol{\alpha}_{j,k}^{f_i}$ are the α-coefficients for the i^{th} edge or face, respectively.
Note that here also internal edges and faces are allowed to deviate from the linear interpolation. This is necessary, if untangling procedures are applied that protrude the curving of the surface into the domain. Further, again analogously to the 2D blending approach, tetrahedral internal bubble shape functions are not necessary to describe the deviation of the element shape from the linear vertex interpolation. They might, however, be advantageous for smoothing the element internal Jacobian after the untangling.

6.3.1 Test Case: Plane Wave Scattering by a Sphere

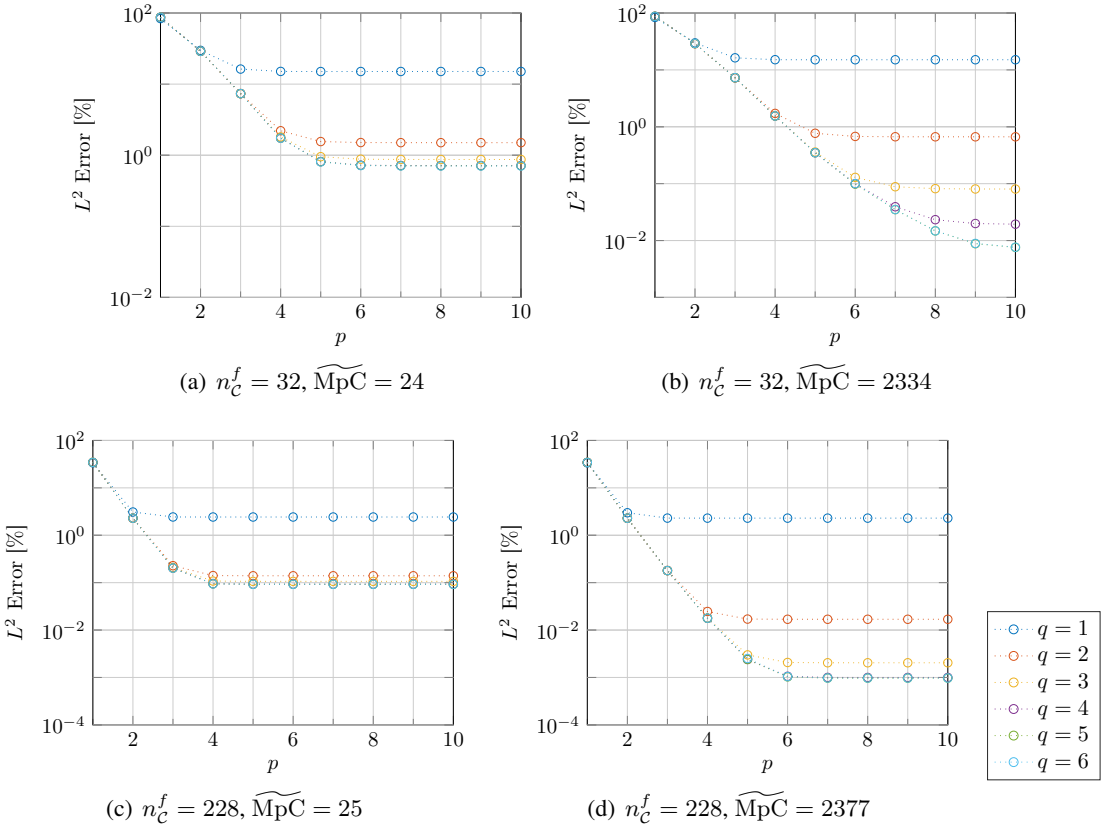
To validate the 3D blending, the scattering of a plane wave by a unit sphere is considered. The computational domain is defined by the scattering unit sphere Γ_s in the centre and a sphere of radius $r_o = 2$ as outer domain boundary Γ_o. This test case is analogous to the scattering by a cylinder from Section 4.4.1. As for the 2D test case, the Helmholtz equation is solved with rigid boundary conditions on Γ_s and characteristic boundary conditions on Γ_o:

$$\begin{aligned} \nabla^2 \mathtt{p} + k^2 \mathtt{p} &= 0, \\ \frac{\partial \mathtt{p}}{\partial n} &= 0 \quad \text{on } \Gamma_s, \\ \frac{\partial \mathtt{p}}{\partial n} &= \frac{\partial \mathtt{p}_{\mathrm{ex}}}{\partial n} + \mathrm{i}\omega \mathtt{p}_{ex} - \mathrm{i}\omega \mathtt{p} \quad \text{on } \Gamma_o. \end{aligned}$$

A derivation of the exact solution $\mathtt{p}_{\mathrm{ex}}$ can be found in [4]. Gauss-Legendre quadratures of order $2p + 2(q-1)$ are applied for the numerical integrations.
For the field error calculation, the relative L^2-error is evaluated over the whole computational domain. The additional error, induced by evaluation points that fall outside of the computational domain and inside the scatterer (see Section 4.4.2, Figure 4.3(a)) is neglected for simplification. The p-convergence curves of the field error for $\omega = \pi$ are shown in Figure 6.8
The curves exhibit the typical behaviour of exponential convergence for lower field orders and stagnation for higher p, with the stagnation level depending on the geometric order q. This validates the blending approach. Further, one can compare the plots with Figures 6.5 and 6.6. There, the p-convergence curves and their stagnation levels correlate well with the results obtained for the GDE_V evaluation towards the Mmesh. For $n_{\mathcal{C}}^f = 32$ and $\widetilde{\mathrm{MpC}} = 2334$ for example, the field error stagnation levels decrease up to order $q = 5$. This corresponds to the GDE_V in Figure 6.5 (b), where the geometry discretisation error also decreases up to $q = 5$. The GDE_V^S in Figure 6.5 (a) on the other hand already stagnated for $q \geq 3$. A similar behaviour can be observed for all depicted cases. As already mentioned above, this indicates that the GDE_V evaluated to the Mmesh is the more reliable measure of the geometric accuracy and justifies the application of this measure in the cylinder test case above.

6.4 Conclusions of this Chapter

This chapter covers the extension of the H^1 modal curving method to 3D. First the theory of the 3D curving is provided, followed by the explanation of the implementation. The latter is divided



(a) $n_{\mathcal{C}}^f = 32$, $\widetilde{\mathrm{MpC}} = 24$ (b) $n_{\mathcal{C}}^f = 32$, $\widetilde{\mathrm{MpC}} = 2334$

(c) $n_{\mathcal{C}}^f = 228$, $\widetilde{\mathrm{MpC}} = 25$ (d) $n_{\mathcal{C}}^f = 228$, $\widetilde{\mathrm{MpC}} = 2377$

Figure 6.8: p-convergence curved of the field error for the plane wave scattering by the unit sphere with $\omega = \pi$.

into the edge curving and the face curving. This separation is needed to ensure continuity between the face elements. The edge curving corresponds to the 2D curving procedure. Therefore, only the detection of the edge target curve has to be outlined. For the face curving, more steps are necessary and explained individually.

The curving implementation is validated with the curving of the sphere. Two GDE measures are applied thereby, one based on the volume to the exact sphere and the other based on the volume to the Mmesh. While the latter behaves as expected, the measure towards the exact sphere shows unanticipated early stagnations. This behaviour could not be explained within the frame of this thesis.

After the general curving implementation, an adaption to allow for feature line treatment is described. It must be noted that this modification does only treat and not detect feature lines, and therefore relies on additional input data. The feature line preservation is verified with a cylinder test case.

Last, the blending approach for the 3D curved mesh is explained and the implementation validated with the scattering of a plane wave by a sphere. Comparison with the field error results and the GDE evaluations in the previous sphere curving test case shows that the GDE evaluated towards the Mmesh is reliable. In retrospect, this also justifies the evaluation of the cylinder test case with this GDE.

Chapter 7

Applications

In this chapter, two applications are provided for the H^1 modal curving algorithm. The first is based on the Munt test case with a relatively simple geometry. It can be seen as the academic generalisation of the second application, where a generic intake of a turbofan engine is considered. The equation for the simulations is the linearised potential equation. In the case of a low Mach number which will be considered in this thesis, it has a structure similar to the Helmholtz equation.

7.1 Acoustics Equations

Linearised Potential Equation

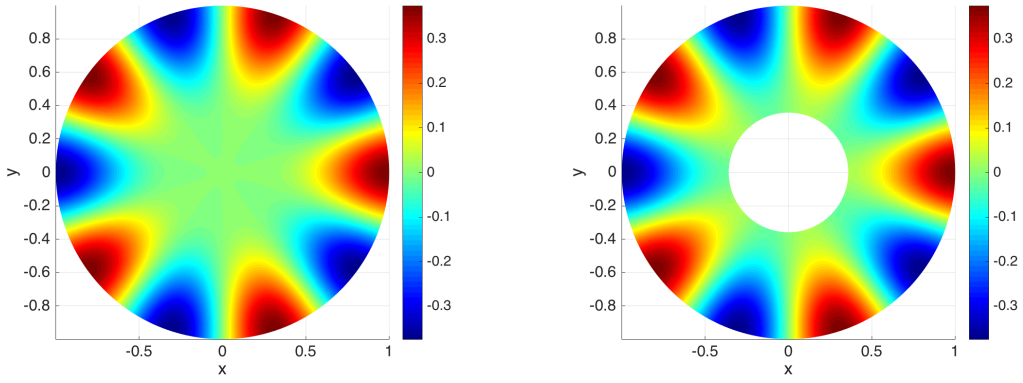
For the application of the modal curving algorithm, simulations with the linearised potential equation are considered. It is the most widely used formula to simulate aircraft noise originating from turbofan engines ([31]).
The equation describes the acoustic propagation in form of the velocity potential ϕ for a steady, potential mean flow of a compressible fluid with constant and uniform entropy. It is given as

$$\rho_0 \frac{D_0}{Dt}\left(\frac{1}{c_0^2}\frac{D_0\phi'}{Dt}\right) - \nabla\cdot\left(\rho_0\nabla\phi'\right) = 0,$$

with c_0 and ρ_0 the mean sound speed and density, respectively, and $\frac{D_0}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{u}_0\cdot\nabla$ the material derivative. The velocity potential ϕ is related to the acoustic parameters, such as the acoustic velocity $\boldsymbol{u}$ and the acoustic pressure $\mathtt{p}$ by $\boldsymbol{u} = \nabla\phi$ and $\mathtt{p} = -\rho_0\frac{D_0\phi}{Dt}$ ([31]).

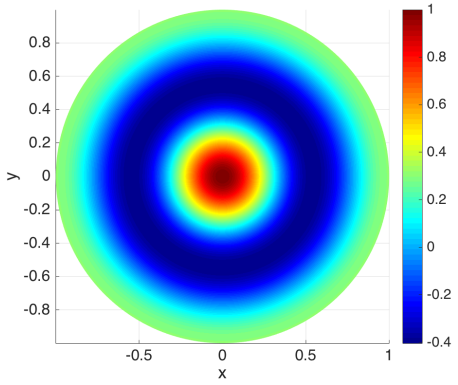
Annular Duct Modes

In both applications, the acoustic field is excited by annular (or cylindrical) duct modes. Modes are often considered in acoustic simulations for ducts with parallel walls, as they are relatively simple solutions of the wave equation that allow for an easier understanding of the acoustic behaviour of the whole simulation than more complicated sound waves. On the other hand, duct modes $\Psi_{\mathtt{m},\mathtt{n}}(x,y)\,\mathrm{e}^{-ik_{\mathtt{m},\mathtt{n}}z}$ form a basis and any solution can be constructed as a linear combination of them ([73]).

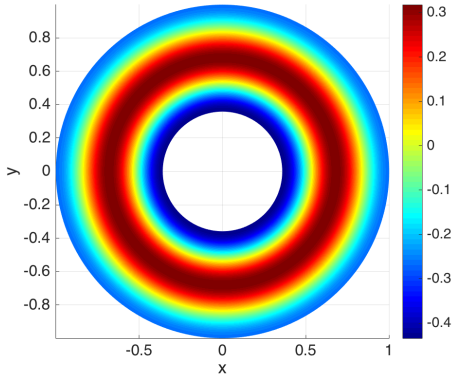


(a) Mode (5,0)

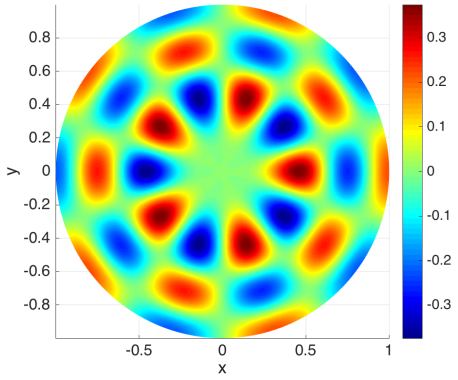
(b) Mode (5,0)



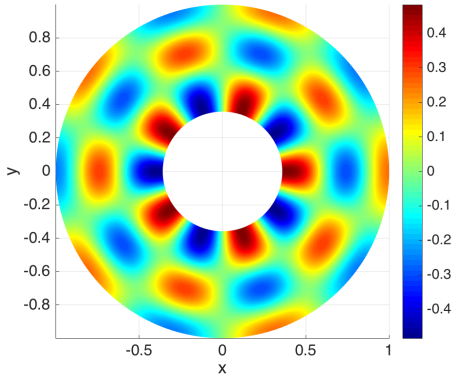
(c) Mode (0,2)



(d) Mode (0,2)



(e) Mode (5,2)

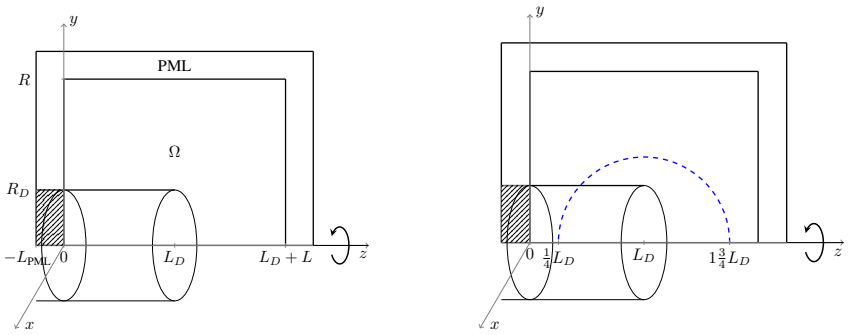


(f) Mode (5,2)

Figure 7.1: Example $\Psi_{\mathtt{m},\mathtt{n}}(x, y)$ at a duct cross section for (a), (c), (e) a cylindrical duct and (b), (d), (f) an annular duct.

For annular and cylindrical ducts of fixed radii, the function $\Psi_{\mathtt{m},\mathtt{n}}(x, y)$ is defined by the azimuthal and the radial mode number ($\mathtt{m}$,$\mathtt{n}$). In Figure 7.1, examples for $\Psi_{\mathtt{m},\mathtt{n}}(x, y)$ are plotted on duct cross sections. The axial wavenumber $k_{\mathtt{m},\mathtt{n}}$ defines the evolution of the duct mode along the axis z and depends on the frequency ω and the fluid and flow parameters. The pressure field has an $\mathrm{e}^{+i\omega t - im\theta}$ dependency on the azimuthal wave number.

Here, an already available code was used to generate the duct modes. This code is based on the derivations in [73].



(a) (b)

Figure 7.2: Figure (a): Schematic of the simulation domain with duct length L_D, duct radius R_D, domain radius R, domain length $L_D + L$, and PML thickness L_{PML}. The shaded area corresponds to the active Pml. The full 3D domain is obtained by rotation around the z-axis.
Figure (b): The blue dashed line indicates the semicircle for the field error evaluation points.

7.2 Modes Exiting a Duct

This application is based on the Munt benchmark problem [62]. It is formed by a semi-infinite cylindrical duct. An acoustic wave is travelling along the duct and diffracted as it exits through the duct opening. The duct walls are infinitelly thin, acoustically impermeable and rigid. Therefore, the boundary condition for the duct wall is a homogeneous Neumann condition for the velocity potential. Along the domain boundary, Sommerfeld radiation condition have to be realised.
In his thesis, Rarata [70] considers the Munt problem to evaluate the effect of varying computational parameters on simulation results with the Discontinuous Galerkin Method. Among these tests, one assesses the influence of the mesh refinement with geometrically linear elements. It is found that the geometric accuracy of the duct description is of high importance for the directivity of the simulation results. With a coarse mesh along the duct, spinning modes get scattered at the C^1-discontinuity between elements and cause high divergence from the analytical solutions at polar angles around $0°$ and $180°$.

7.2.1 Simulation Parameters

For this application, the elements defining the duct wall are curved using the H^1 modal curving approach, presented in Section 6. The effect of the curved mesh on the accuracy of the simulation results is considered.
A sketch of the simulation domain Ω is provided in Figure 7.2(a). The domain parameters are defined in terms of the duct radius R_D as $L_D = 2R_D$, $R = 3R_D$, and $L = 2R_D$. For simplicity, the duct radius is chosen as $R_D = 1$. In order to fulfill the Sommerfeld radiation condition, a PML is added to the simulation domain. Two Mmesh refinements are considered, a relatively coarse Mmesh with $\widetilde{\text{MpC}} = 47.75$ and a fine Mmesh with $\widetilde{\text{MpC}} = 2753.0$.
To implement an infinitely thin, impermeable duct, it is generated as a two-layered surface. In the mesh generation, the two layers are separated by increasing the radius of the outer layer up to

$R_D + \varepsilon$ at the back end of the duct, with $\varepsilon = 0.1$. At the open end, both layers come together in a single circle at the predefined duct radius R_D. After the linear mesh generation and before the curving, the outer layer is moved back to the duct radius R_D over the whole duct length. Working with two layers involves the risk of intersecting mesh surfaces, but as they are treated as separate boundary domains, this will not affect the desired impermeability property. For this application case, only the two duct layers are curved. Afterwards, it has to be ensured that the mesh remains continuous along the feature lines between the curved surfaces and their adjacent boundary domains. Therefore, elements on neighbouring surfaces that are adjacent to the curved surface are adjusted to follow the curving of the edges on those feature lines.
Circular duct modes are inserted at the back wall of the duct at $z = -L_{\text{PML}}$, $\sqrt{x^2 + y^2} \leq R_D$. The intersection between the PML and the duct is defined as a so-called *active PML*. There, only the backtravelling component of the acoustic field is damped, while the forward moving mode is actively prescribed [38]. Implementations of the passive and the active PML were already available and are not newly implemented in the frame of this thesis.
Compared to [70], the problem size is reduced by considering modes with $kR_D = \omega = 4$ and $kR_D = \omega = 6$. The mesh length is kept fixed for both cases and corresponds to $\frac{1}{2}$ or $\frac{1}{3}$ element per wavelength, respectively.
For the simulation, an existing implementation of the adaptive high-order FEM is used [31]. The target accuracy for the a-priori error indicator is selected as 5%, with a maximal element field order of $p = 15$.
Following the test case in [70], no flow is applied on the simulation domain, as no additional effect is expected on the geometric accuracy dependency from adding flow to the simulation.

7.2.2 Error Evaluation

For the calculation of the GDE, the volume based measure GDE_V described in Section 6.1.4 is applied. Only the duct is curved, and therefore the GDE is only evaluated over the two layers of the duct surface.
The simulation results are evaluated along a semicircle in the simulation domain (see Fig.7.2(b)). Along this curve, the potential field is obtained by interpolation and then used to calculate the corresponding pressure. The latter is transformed into the *sound pressure level* (SPL) by

$$L_{\mathrm{p}} = 10 \log_{10} \frac{\mathrm{p}_{\text{rms}}^2}{\mathrm{p}_{\text{ref}}^2},$$

with the root-mean-square pressure value $\mathrm{p}_{\text{rms}} = |\mathrm{p}|^2/2$ and the standard reference pressure of $2 \cdot 10^{-5}$Pa (see [73]).
For this academic application case, an analytical solution is available. The derivation and calculation are outlined in [30]. Here, the analytical solution was provided by G. Gabard.

7.2.3 Results

In Figure 7.3, the GDE development is shown for the two considered Mmesh refinements. For both cases, there is a strong improvement of the geometric accuracy from linear to quadratic meshes. With the latter, a significantly better geometric description is obtained with the finer

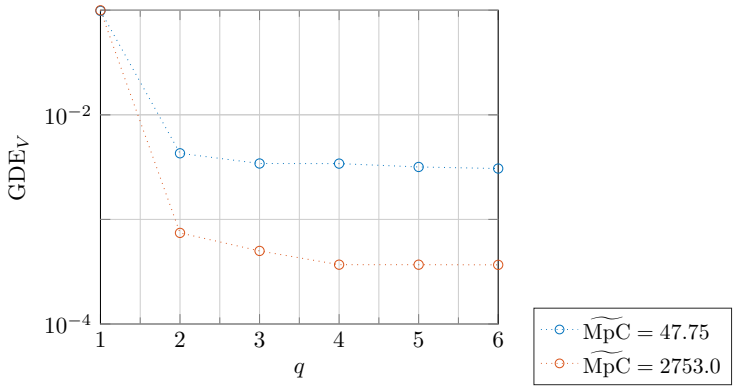


Figure 7.3: GDE curves for the modes exiting a duct application case.

Mmesh. Further, the GDE curve for $\widetilde{\mathrm{MpC}} = 47.75$ stagnates already from $q = 2$, while the geometric accuracy for $\widetilde{\mathrm{MpC}} = 2753.0$ improves up to order $q = 4$. This corresponds to the expected behaviour of a later stagnation onset on a lower stagnation level with the finer Mmesh. It is also similar to what was observed in the previous chapters.

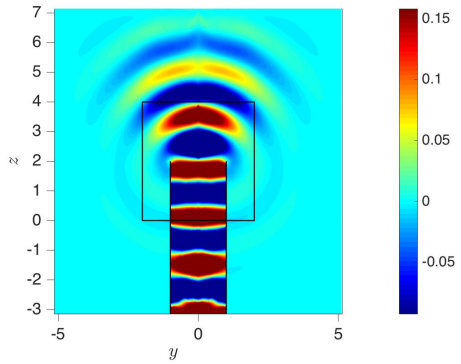
As mentioned above, two frequencies are considered. For $\omega = 6$, simulations with the fully cut-on plane wave $(0, 0)$, the intermedially cut-on mode $(2, 0)$, and the close to cut-off mode $(4, 0)$ are evaluated. Additionally, the simulations of the plane wave are obtained for $\omega = 4$. Figure 7.4 depicts numerically obtained potential fields of these two cases.

The sound pressure levels of the simulation results and of the analytical solution are plotted in Figures 7.5 and 7.6. The plots 7.5(a) and 7.5(b) for $\omega = 6$, mode $(0, 0)$ are dominated by a strong deviation of the simulation results from the analytical solution, close to the duct wall. Curved meshes do not resolve or even improve this issue, but rather lead to a generally worse match between the numerical and the analytical SPL. Within the time frame of this thesis, no explanation was found for this behaviour. It also comes together with an effect that looks like a strong backtravelling wave in the active PML domain (see Fig. 7.4(b)).

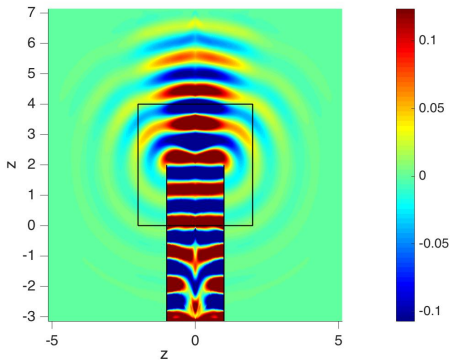
With increasing azimuthal mode number, the described effect diminishes (compare plots 7.5(c) - (f) for modes $(2, 0)$ and $(4, 0)$). It is also not significant for the plane wave at the lower frequency $\omega = 4$. There, a clear improvement is observed when changing from a linear to a curved mesh. Among the latter, no notable improvement can be observed between the simulation results on meshes with different curving orders.

For the simulation with $\omega = 6$, mode $(4, 0)$, $\widetilde{\mathrm{MpC}} = 47.75$, and $q = 1$, a directivity error is present at polar angles around $0°$ and $180°$, similarly as described in [70]. On curved meshes, this error is reduced by up to $20dB$ at $\Theta = 0°$ with order $q = 3$, and up to $40dB$ in the duct interior at $\Theta = 180°$, for $q = 4$. As the SPL is a logarithmic measure, this corresponds to a reduction of the sound pressure by one or two orders of magnitude, respectively. The effect that the best simulation results were not obtained with the highest tested curving order $q = 6$ might be explained by the fact that the curving target is relatively coarse and therefore includes some inaccuracy itself.

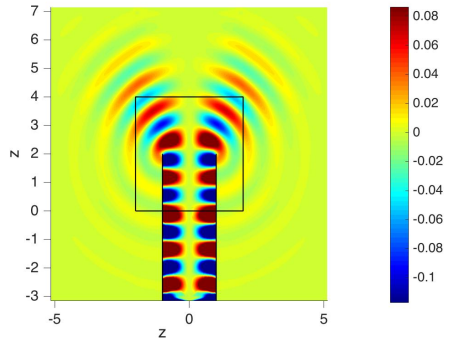
However, with the more refined Mmesh, the above described directivity error is only visible in the duct interior. An explanation for the differences in the simulation results on the two linear, and therefore uncurved and otherwise untreated Cmeshes, lies in the preprocessing step that



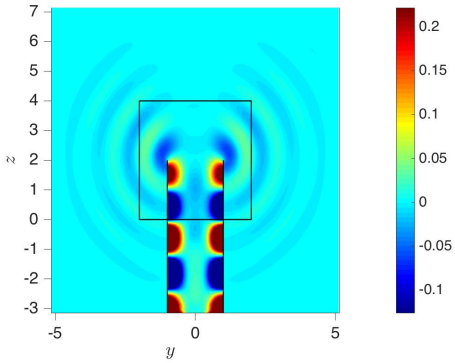
(a) $\omega = 4$, mode $(0, 0)$



(b) $\omega = 6$, mode $(0, 0)$

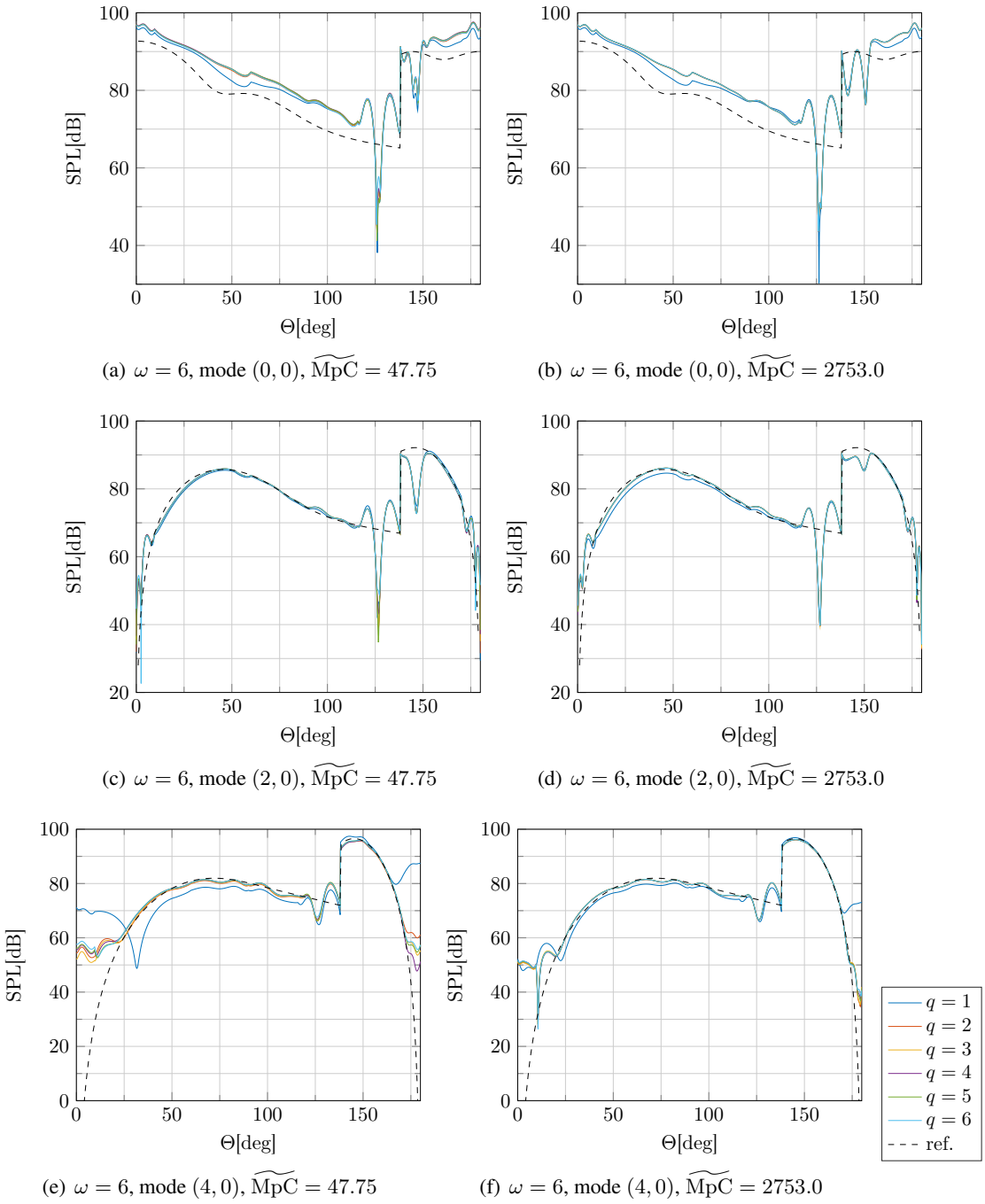


(c) $\omega = 6$, mode $(2, 0)$



(d) $\omega = 6$, mode $(4, 0)$

Figure 7.4: Plots of the potential field of the modes exiting a duct application case, obtained from simulations with $\widetilde{\mathrm{MpC}} = 2753.0$. The black box indicates the boundary between the simulation domain and the PML.



(a) $\omega = 6$, mode $(0, 0)$, $\widetilde{\mathrm{MpC}} = 47.75$

(b) $\omega = 6$, mode $(0, 0)$, $\widetilde{\mathrm{MpC}} = 2753.0$

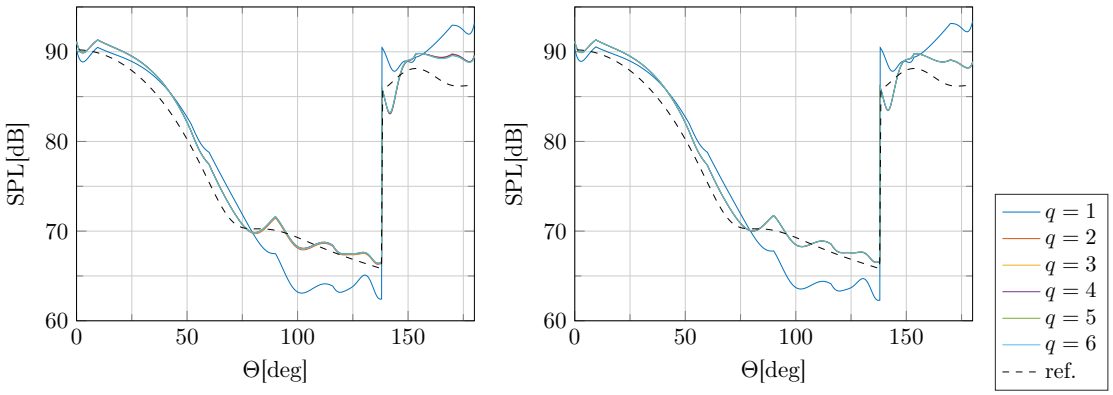
(c) $\omega = 6$, mode $(2, 0)$, $\widetilde{\mathrm{MpC}} = 47.75$

(d) $\omega = 6$, mode $(2, 0)$, $\widetilde{\mathrm{MpC}} = 2753.0$

(e) $\omega = 6$, mode $(4, 0)$, $\widetilde{\mathrm{MpC}} = 47.75$

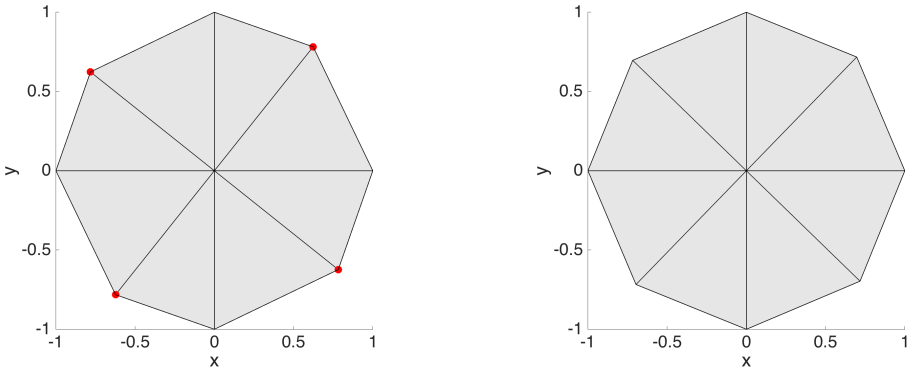
(f) $\omega = 6$, mode $(4, 0)$, $\widetilde{\mathrm{MpC}} = 2753.0$

Figure 7.5: Sound pressure levels of the modes exiting a duct application case, evaluated along a semicircle in the computational domain.



(a) $\omega = 4$, mode $(0, 0)$, $\widetilde{\mathrm{MpC}} = 47.75$ (b) $\omega = 4$, mode $(0, 0)$, $\widetilde{\mathrm{MpC}} = 2753.0$

Figure 7.6: Sound pressure levels of the modes exiting a duct application case, evaluated along a semicircle in the computational domain.



(a) $\widetilde{\mathrm{MpC}} = 47.75$ (b) $\widetilde{\mathrm{MpC}} = 2753.0$

Figure 7.7: Cmesh triangulation of the duct cross-section at $z = 0$. The circumferential line describes the duct wall. In (a), the red dots highlight the Cmesh nodes that were moved a relative big distance in the preprocessing step to align them with Mmesh nodes.

moves the Cmesh nodes to the closest Mmesh nodes. Recall that this step is necessary in our setup to ensure that the Cmesh nodes form a subset of the Mmesh nodes, a prerequisite we formulated for our curving routines.
In Figure 7.7, a cross section of the linear Cmesh in the duct at the mode insertion plane is plotted for both Mmesh refinements. With the fine Mmesh, the Cmesh is still visually equally sized. In the other case, the Cmesh nodes had to be moved larger distances to align with nodes on the coarse Mmesh. Therefore, the resulting linear Cmesh is not equisized anymore, but still displays some regularity, due to the general mesh generation along the circular duct. For the simulation results, this seems to lead to a stronger scattering of the mode along the duct wall, and overall to a larger directivity error.

In summary, the modes exiting a duct application case comes with several overlying effects that are only partially understood and explained here. But for the cases where the directivity error is dominating in the linear case, curved meshes lead to an improvement of the simulation results. In these instances, the reduction of the SPL in the critical domains is up to a similar extent as it is obtained with mesh refinement in [70], especially in the duct interior at $\Theta = 180°$. The biggest

effect is thereby obtained when changing from linear meshes to meshes with curving order $q = 2$. Higher curving orders lead in some cases only to small (or no visual) improvements.

7.3 Modes Exiting a Generic Intake

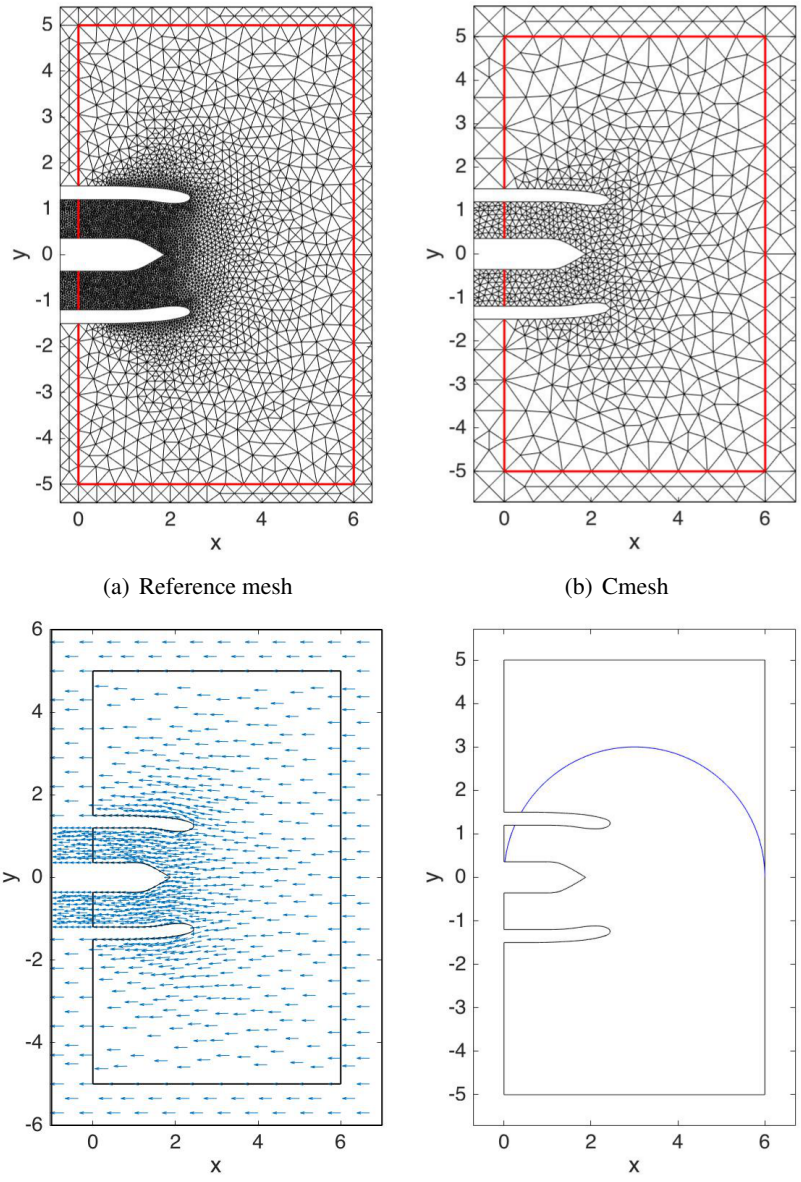
The previous problem can be seen as an academic simplification for a turbofan intake. In this application, a more realistic, generic geometry of an intake is considered. First, a 2D version is curved and used as simulation domain. Afterwards, a 3D geometry is obtained by rotation around the x-axis. The 3D curving algorithm is applied and the geometric accuracy is evaluated. However, simulations can not be run on the resulting domain mesh due to tangled elements.

7.3.1 2D Geometry

Plots of the geometry and the Cmesh are shown in Figure 7.8. The generic intake is defined by an upper and lower part of the nacelle and the spinner in the middle. It is based on the generic intake geometry described in [31]. A PML encloses the simulation domain to ensure compliance with the Sommerfeld radiation condition. The nacelle and the spinner are simulated as rigid walls by applying homogeneous Neumann boundary conditions on the velocity potential.
Between the nacelle and the spinner, the lines defining the fan plane are located at $x = 0$. The velocity field is excited in the form of annular duct modes at the back of the PML behind these lines. In between the insertion and the fan line, an active PML is implemented to ensure the propagation of the wave.
As this application is a 2D simulation, only modes with azimuthal order 0 are considered. For the flow conditions, an approach case is considered (see Fig. 7.8(c)). An already available data set (see [31]) is scaled to obtain a mean sound speed and a mean density of 1. Simulations are run at $\omega = 20$ for the plane wave $(0, 0)$, an intermediately cut-on mode $(0, 2)$ and a nearly cut-off mode $(0, 5)$. Varying MpC-refinements are considered, but as they do not lead to any additional insight, only the results for $\mathrm{MpC} = 1000$ are shown below.
For this geometry, there is no analytical solution available. Instead, a numerical reference solution is obtain on a refined mesh (see Fig. 7.8(a)), nodally curved with $q = 2$ and field order $p = 15$. Plots of the reference potential fields are shown in Figure 7.9. Analogously to the modes exiting a duct application case, the SPL is evaluated along a semicircle (see Fig. 7.8(d)). Note that the evaluation arc is crossing the nacelle. For this reason, the resulting SPL plots below have a gap as well.

Results

In Figure 7.10, the evolution of the GDE_A is shown for four different MpC-refinements. With $\mathrm{MpC} = 100$, the coarsest refinement, the geometric accuracy significantly increases from $q = 1$ to $q = 2$. An additional increase of q improves the geometric accuracy only slightly. The curves of the remaining refinements fall together and converge over the full curving order range $q = 1 - 6$. This indicates that from $\mathrm{MpC} = 200$ on, the inaccuracy of the target mesh description is negligible for the GDE_A calculation.



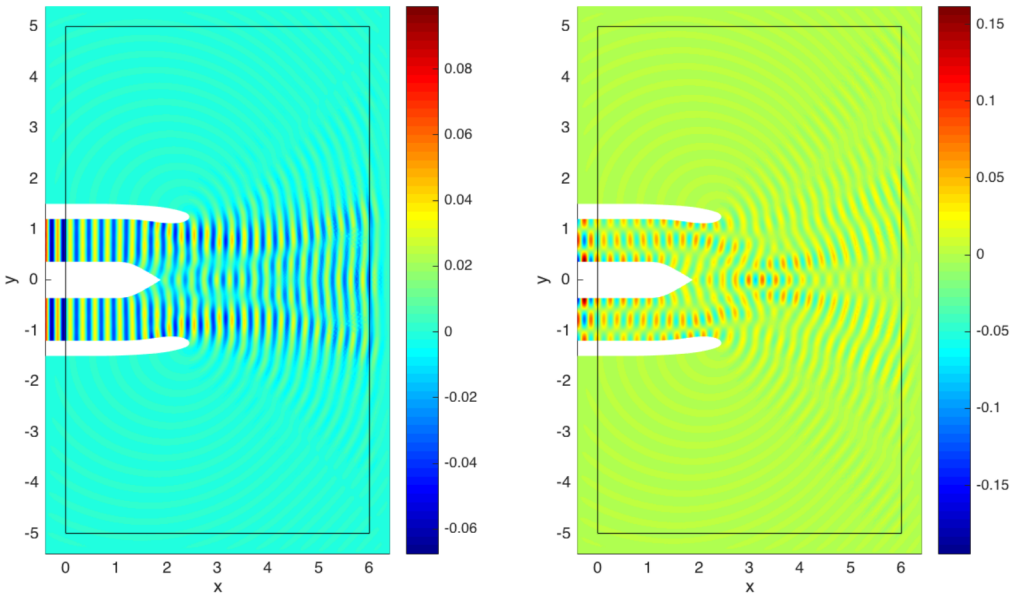
(a) Reference mesh

(b) Cmesh

(c) Flow data

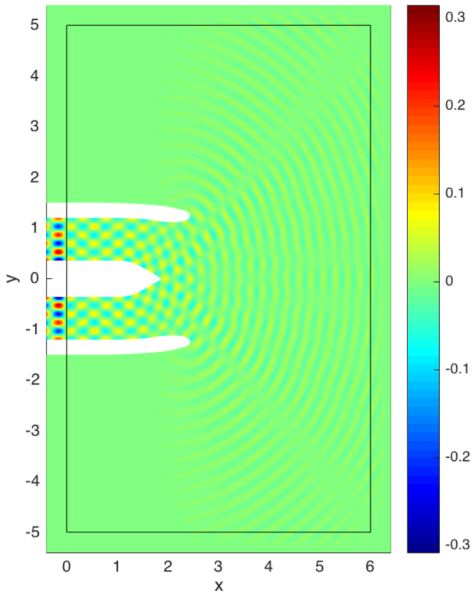
(d) Evaluation points

Figure 7.8: Plots of the 2D geometry and flow data of the generic intake application case. In (a), the mesh for the reference simulation is shown, while (b) depicts the Cmesh. The boundary that separates the simulation domain from the PML is highlighted in red. Figure (c) depicts the flow applied for the simulation, and the blue semicircle in Figure (d) displays the position of the SPL evaluation points.



(a) Mode (0,0)

(b) Mode (0,2)



(c) Mode (0,5)

Figure 7.9: Reference potential field for $\omega = 20$ of the 2D generic intake application case, obtained by numerical simulations.

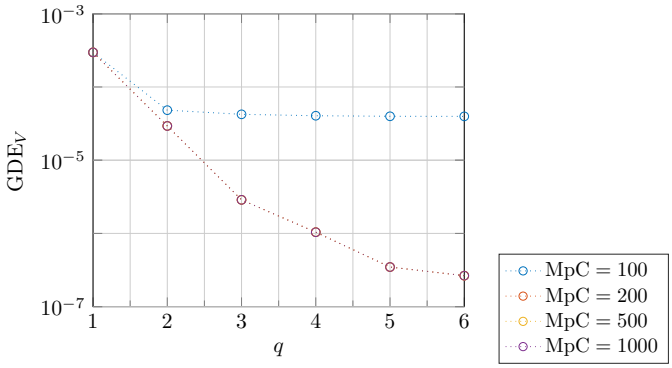
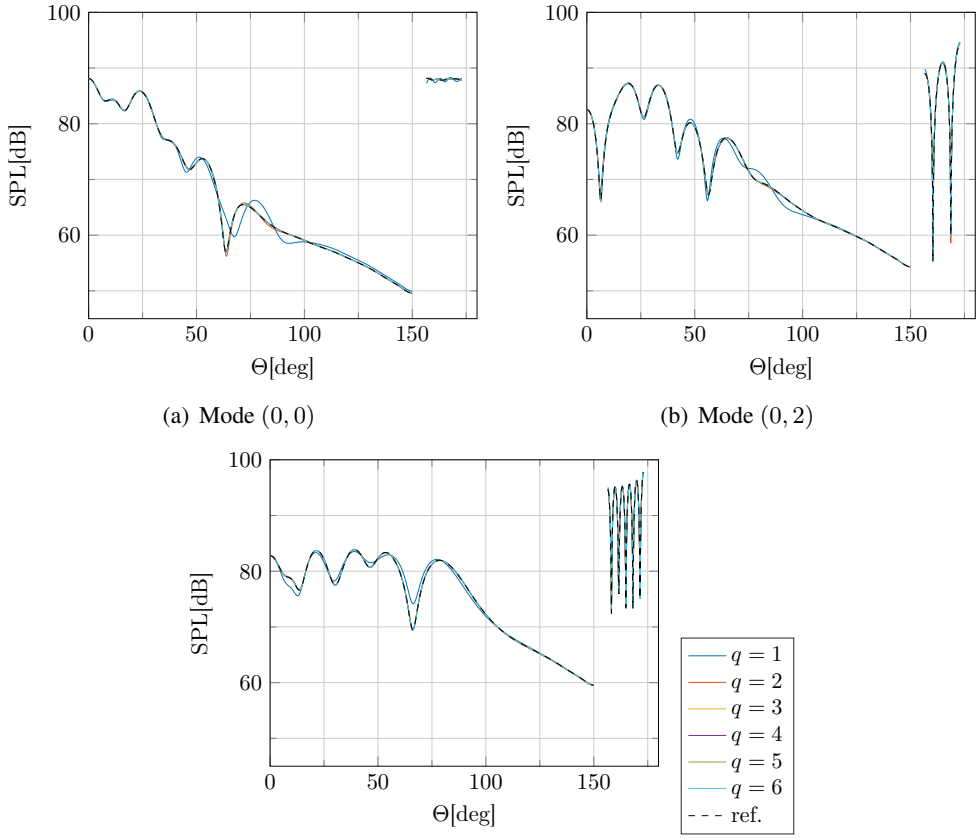


Figure 7.10: Area-based GDE of the meshes for the 2D generic intake application case.



(a) Mode $(0, 0)$

(b) Mode $(0, 2)$

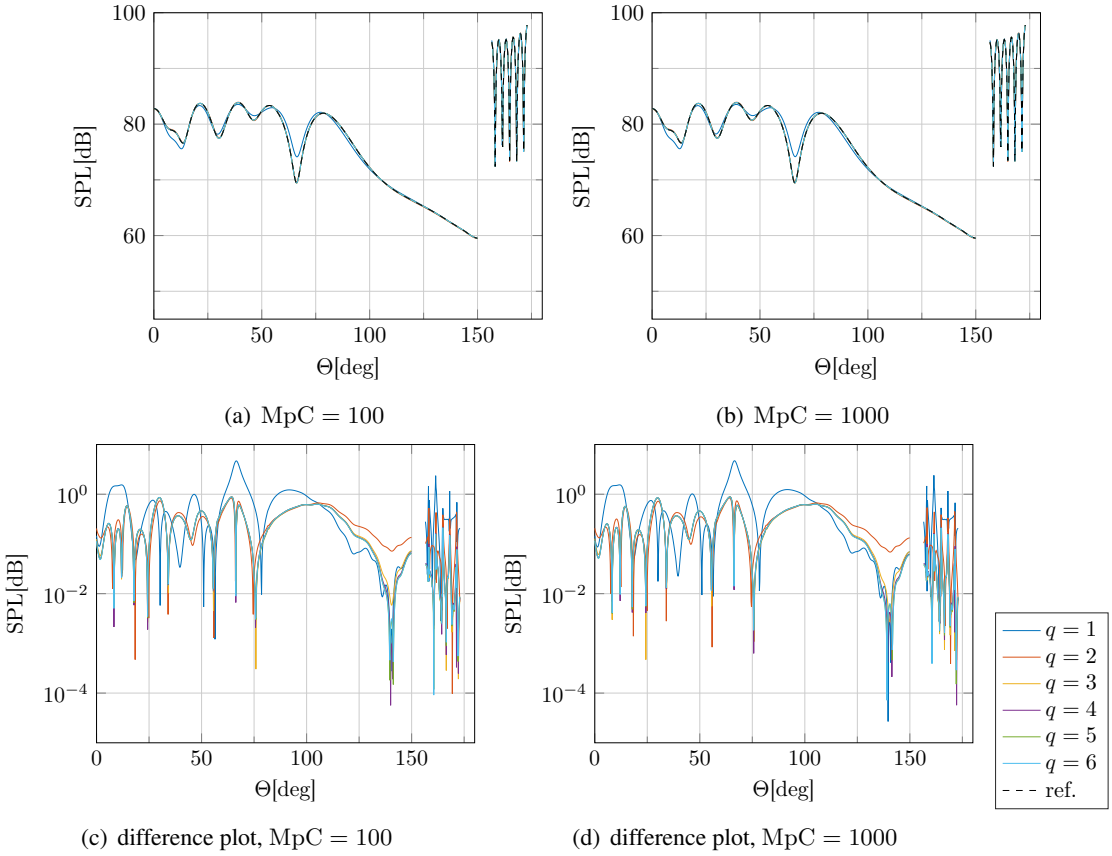
(c) Mode $(0, 5)$

Figure 7.11: Sound pressure levels of the 2D generic intake application case with $\omega = 20$ and $\mathrm{MpC} = 1000$, evaluated along a semicircle in the computational domain.

The sound pressure levels of the simulation results are displayed in Figure 7.11 for the three considered modes and simulations with $\mathrm{MpC} = 1000$. Despite the alike GDE_A results for $\mathrm{MpC} = 200$, $\mathrm{MpC} = 500$, and $\mathrm{MpC} = 1000$, the highest MpC-refinement is considered because of the conclusion in Chapter 5 that the GDE_A does not cover all significant contributions. For all considered modes, the curving in general from $q = 1$ to $q = 2$ leads to an improvement of the evaluated SPL. This is especially the case for the plane wave $(0, 0)$, where a directional shift in the solution on meshes with $q = 1$ is resolved on the curved meshes. The curves for curving orders $q = 2$ to $q = 6$ lie too close together to allow a visual comparison.
Similarly, the plots for mode $(0, 5)$ with $\mathrm{MpC} = 100$ and $\mathrm{MpC} = 1000$ in Figures 7.12(a) and 7.12(b), respectively, do not reveal any significant difference. Therefore, the corresponding error in the SPL is shown in Figures 7.12(c) and 7.12(d), in form of the absolute difference between the simulation result SPL and the reference SPL. However, besides the already identified higher error with $q = 1$, no clear tendency is observed among the higher curving orders. Also, no improvement can be detected when changing from $\mathrm{MpC} = 100$ to $\mathrm{MpC} = 1000$. The plots for the remaining modes and MpC-refinements imply the same conclusions.

7.3.2 3D Geometry

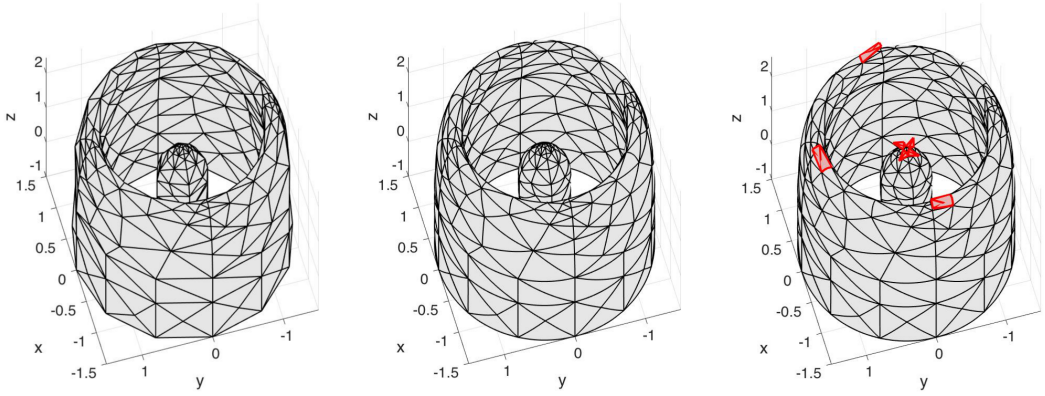
This section is a short presentation of curving results for the 3D version of the generic intake from the previous section.



(a) MpC = 100 (b) MpC = 1000

(c) difference plot, MpC = 100 (d) difference plot, MpC = 1000

Figure 7.12: Sound pressure levels of the 2D generic intake application case with $\omega = 20$ and mode $(0, 5)$, evaluated along a semicircle in the computational domain. In (c) and (d), the absolute difference to the reference SPL is plotted.



(a) $q = 1$ (b) $q = 6$ (c) $q = 6$, tangled elements

Figure 7.13: Plot of the (a) linear and (b) curved Cmesh of the 3D generic intake. In (c), the tangled elements are added in red. For clarity, only the nacelle and the spinner are shown.

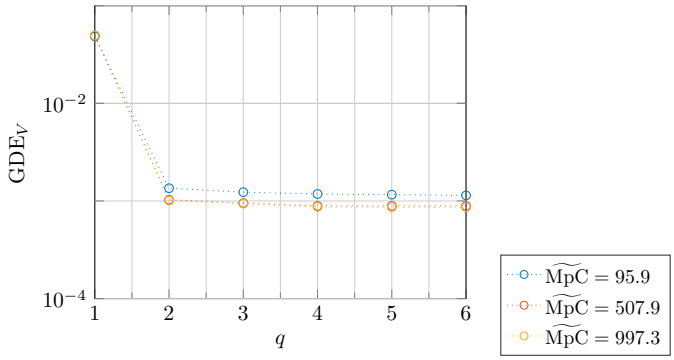


Figure 7.14: Area-based GDE of the meshes for the 3D generic intake application case.

The 3D generic intake geometry is obtained by rotating the upper half of the 2D geometry around the z-axis. The 3D curving approach is applied on the spinner and the nacelle surface. In Figure 7.13, the linear Cmesh and a curved version with $q = 6$ are shown.
Since only the nacelle and the spinner are actually curved, the GDE is also evaluated based only on those two surfaces. It is plotted in Figure 7.14 for three different $\widetilde{\mathrm{MpC}}$-refinements. For increasing geometric order, the GDE_V decreases well up to $q = 2$, before it nearly stagnates.
As mentioned at the beginning of this Section, no simulations could be applied to the curved geometries because of tangled elements (see Fig. 7.13(c)). An untangling routine as those described in Section 2.4 would be needed as a postprocessing step in order to allow for reliable simulations.

7.4 Conclusions of this Chapter

This chapter provides application cases for the H^1 modal curving algorithm described in the previous chapters. The first application is based on the Munt test case. For this case, the dependency of the solution on the geometric accuracy is known ([70]). While in [70] the geometric description is improved by mesh refinement, a coarse mesh is maintained here. Relative to the wavelength, the mesh refinement considered here is even coarser than the coarsest mesh in [70].

In the simulation results for some modes, additional effects overlay the expected geometry dependent error. These effects are only partially understood and explained in this thesis. But for some cases, the geometric dependent error is observed with the linear mesh. There, mesh curving leads to improvements in the solutions up to a similar extent as in [70]. Remarkable is that the strongest improvement from curving order $q = 1$ to $q = 2$, and from $q = 4$ on, no significant improvement is observed in any case.
This first application can be seen as the academic generalistion of the second application, an generic intake. For this more realistic geometry, the curving of the 3D version comes with the issue of tangled elements. In this thesis, no untangling scheme is considered, therefore only the geometric accuracy is evaluated. It reveals a clear decrease of the GDE_V from $q = 1$ to $q = 2$. Only minor improvements are obtained with higher curving orders.
A 2D version of the geometry is considered as well, which allows for simulations, as the curving does not lead to tangled elements. There, again a visual improvement is obtained when changing from a linear to a curved mesh with $q = 2$. While the GDE_A improves, dependent on the MpC-refinement, over the whole considered range $q = 1 - 6$, no significant improvements are observed for the simiulation results with higher curving orders.
Overall, curved meshes in general prove advantageous for the two applications with the linearised potential equation. However, the 2D generic intake case suggests that the positive effect is weaker than for Helmholtz simulations, especially when increasing the curving order $q > 2$. As here only meshes are considered that are curved with the H^1 modal approach, no comparison with respect to other, e.g. nodally curved meshes can be made. It is observed that the mesh curving is only meaningful up to a certain, problem dependent order, from which on only marginal or no significant improvement of the simulation accuracy is obtained. Further, the 3D version of the generic intake highlights the need for an untangling postprocessing step.

Chapter 8

Conclusion and Future Work

This chapter summarises the thesis and the resulting conclusions. Open questions are identified and possible future work outlined.

8.1 Summary and Conclusions

This thesis investigates the improvement of p-FEM simulations in numerical acoustics by high-order geometric enhancement. It is built up of two main objectives. The first is the creation of meshes with curved boundaries with only a fine linear target mesh as geometry description. The second objective is the analysis of the influence of the geometric accuracy of the mesh on the simulation results for acoustic simulations, specifically Helmholtz scattering.

In Chapter 2, a review of different topics concerning geometric enhancement is given. In the conclusions of that chapter, the main decisions on the content of this thesis are discussed based on the literature and the assumptions on the available geometry data.
Polynomial interpolation with the standard, nodal Lagrange polynomials and the modal Lobatto polynomials are chosen to be further considered for the geometric representation. Available mesh curving algorithms are classified into three groups. For the two main classes with either full knowledge of the exact geometry, or no knowledge besides a coarse linear mesh, several methods are available in literature. Besides quadratic fittings, one high-order curving approach was found that belongs to the same class as the methods in this thesis. The difference between this approach and the ones developed here lies mainly in the choice of modal shape functions for the geometric representation.
The review on GDE measures and on untangling procedures shows that for both there is already a range of methods or measures available. From literature alone, no preferences could be found from the discussed measures of geometric accuracy. Therefore, several measures were selected to be applied and tested within the thesis. It is further decided not to consider or develop any untangling or smoothing routine.

Four mesh curving methods are developed that take a fine target mesh as geometric description: a nodal curving algorithm, a least squares modal approach, a H^1 modal curving, and a reference-based modal curving algorithm. The modal methods are built by a coupling of

established components such as the least squares approach and in the meshing community rarely considered modal Lobatto polynomials. For the H^1-optimisation that is motivated by their definition and structure, no reference was found in the literature on mesh curving.
The methods are described in Chapter 3 and compared with two test cases, the unit circle and a distorted ellipse. For the circle, the nodal curving implemented in *Gmsh* ([36]) is considered as reference that has the exact geometry as target curve. The geometric accuracy is evaluated with an area-based measure, similar to that introduced in the context of the reference algorithm [72]. Based on the literature review, this is the first comparison between nodal and modal curving methods.
Two of the four methods are dismissed at the end of Chapter 3. The nodal curving method suffers from convergence issues within the curving steps, which is why the whole approach is found to be too unstable. A short test with alternative node spacings was performed but did not result in any improvement. The second method that is rejected is the reference-based modal approach. This is justified by a weaker convergence compared with the other three methods for the circle test case and restrictions in the choice of mesh sizes for the distorted ellipse.
Between the other approaches, no clear preference is obtained from the two test cases. The results of the H^1 modal based algorithm are limited by the accuracy of the underlying linear target mesh. This is not the case for the least squares approach, which does not rely on the interpolated data. Instead, the method considers only geometrically exact nodes of the target mesh. However, the least squares curving is unstable or even uncalculable for too coarse target meshes. For both curving methods, modifications could be considered to bypass their drawbacks (see Sec. 8.2). In this thesis, only the described basic approaches are considered.

At the beginning of Chapter 4, a short review of numerical acoustics, as well as high-order geometric implementations in FEM is provided. For the modally curved meshes, a blending approach is developed to embed the curved boundary elements into the simulation domain. The resulting meshes are applied for the 2D scattering test case of a plane wave by an infinite cylinder. The obtained simulation results show the dependency of the field error on the geometric accuracy of the domain description. Once the field discretisation is fine enough, the geometric inaccuracy of the simulation domain becomes the dominating factor of the field error. Further, some of the results indicate that meshes that were curved with the H^1 modal algorithm can lead to a lower field error than the corresponding meshes that were curved with the least squares approach.

Motivated by these results, the relation between the geometry discretisation error (GDE) and the geometry induced error (GIE) is evaluated in Chapter 5. Therefore, again the scattering of a plane wave by an infinite cylinder is considered with nodally curved meshes that were obtained using *Gmsh*. While in the literature only comparative statements were found on the GDE-GIE relation, this thesis provides a quantitative study for the Helmholtz equation. Three GDE measures are tested, as well as three field error measures. For all reasonable combinations of these measures, a linear dependency of the GIE with the GDE, and a super-linear dependency with the frequency ω is found.
Next, the model of the relation between the area based GDE and the GIE evaluated along a ring in the computational domain is tested for modally curved meshes. As it already foreshadowed in

the previous chapter, the novel H^1 modally curved meshes prove favourable for the acoustic simulation, compared to the meshes curved with the least squares approach. However, the advantageous property of the H^1 modally curved meshes is not explained by the considered GDE measure. This is further discussed as future work in Section 8.2.

In Chapter 6, the H^1 modal curving approach is extended to 3D. To ensure continuity in between face elements, the curving step is separated into edge and face curving. The former coincides with the 2D curving algorithm, and only the detecion of the target curves needs to be adapted to 3D. For the face curving, more steps are necessary to define the target of the curving. After the detection of the target points they are triangulated and then a parametrisation is formed to define the mapping from the reference element.
The curving is extended to also enable the treatment of feature lines. As this thesis does not cover the detection of feature lines, they are included as an additional input along with the coarse and the fine linear mesh, in form of boundary groups that are bounded by the feature lines. This limits the feature lines to closed loops on the surface, or curves that devide the domain boundary in two halfes. Other ways are possible to define feature lines, as well as the coupling with a feature line detection algorithm as preprocessing step (see Sec. 8.2).
The extension of the blending approach to 3D is described to allow for simulations on 3D domains with curved boundaries.
Two test cases are considered to validate the curving and the feature line treatment, the unit sphere and a cylinder. The GDE is measured with a measure based on the volume between the curved mesh and the target. For the sphere, an additional measure based on the volume between the curved mesh and the exact sphere is considered. The latter measure does not show the expected behaviour and the reasoning therefore could not be found in the scope of this thesis. This is left for future work (see Sec. 8.2). However, the test case of a plane wave scattered by the unit sphere confirmes that the GDE evaluated towards the Mmesh is a reliable measure. This test case also validates the embedding of the curved boundary into the domain mesh.

In Chapter 7, two applications are described with meshes curved by the H^1 modal curving algorithm, one based on the academic Munt test case, the other on a generic geometry of an intake. For the latter, both a 2D and 3D version is considered, as the curved versions of the 3D mesh suffer from tangled elements and therefore do not allow reliable simulations. The applications are simulations with the linearised potential equation.
Where no other effects are superimposed, a clear increase of the simulation accuracy is obtained with the curved meshes. In all cases, the biggest improvement is obtained when comparing the linear mesh with curved meshes with $q = 2$. For the 2D generic intake, higher curving orders lead to an improvement of GDE_A, but not in the simulation results. The GDE_V for the 3D generic intake on the other hand decreases well down to $q = 4$. For the Munt test case, the optimal curving order is $q = 2$, $q = 3$, or $q = 4$, depending on the $\widetilde{\mathrm{MpC}}$-refinement and the considered mode.
Thus, the general positive effect of curved meshes on the simulation accuracy is confirmed also for LPE simulations. The 2D generic intake test case though suggest that this effect is smaller than for the Helmholtz simulations. Also the advantageous properties of the H^1 modal curving

seen in Chapter 4, where an improvement of the simulation accuracy is obtained even with stagnating GDE_A is not observed. Further, the 3D generic intake case reinforces the need for an untangling routine as a postprocessing step after the curving.

8.2 Future Work

Not all possible paths could be considered within this thesis. Throughout the text, remarks were added to indicate where further research or improvements could be applied. These open issues are summarised and discussed in this section.

8.2.1 Improvements of Curving Methods

In this thesis, basic versions of the different curving methods are considered and compared. Especially for the two more promising approaches, the modal least squares and the H^1 modal curving, their performance could be most likely improved by some modifications or preprocessing steps.

For the least squares approach, two ways to bypass its instabilities were suggested by one of the reviewers of [99]. One is to relax the condition of using the exact Mmesh nodes in favor of an optimised node distribution, similarly to what is tried for the nodal curving approach in Section 3.3.2.2. The other suggestion is to apply other polynomials for the interpolation, such as Legendre polynomials with constraints.

The main drawback of the H^1 modal method lies in the dependency of the Mmesh accuracy. This bottleneck could be avoided by improving the geometric accuracy of the Mmesh in a preprocessing step with a surface reconstruction method. Such a preprocessing step is e.g. applied in the algorithm described in [46]. In Section 2.3.2, a short summary is provided of literature on surface reconstruction methods.

For the described 3D development of the chosen H^1 modal approach, further improvements could be considered. First, the feature line definition applied here could be improved by directly marking individual edges as feature lines. This also allows feature lines that are not bounding a closed boundary domain but end in the middle of the surface. Further, a feature line detection algorithm could be implemented as a preprocessing step to override the current requirements of additional input information for the feature line treatment. Such algorithms are available in the literature, e.g. [47].

High priority should be assigned to adding an untangling method as a postprocessing step of the curving. This is necessary to apply meshes curved with the described methods to simulations with real life 3D geometries. As mentioned in Chapter 2, there is a lot of work done on this topic. An untangling routine that seems appropriate to combine with the H^1 modal curving is the one developed by Gargallo et al. [34, 33], as it does not rely on a specific polynomial basis for the geometric description.

Finally, the performance of the H^1 modal curving algorithm could be improved by taking advantage of the hierarchic structure of the Lobatto polynomial bases. Similar to the adaptive

choice of the field order, the curving order of each boundary element could be chosen individually with only small constraints based on the directly neighbouring elements. In flat regions of the geometry, the curving order would be set to $q = 1$, while it would be increased in regions of high curvature. Combined with the above mentioned untangling procedure, the curving order could be reduced in regions where tangling is not resolved, until a curving order is reached that could be tackled by the untangling method.

8.2.2 Improvements of the GDE Evaluation

The evaluation of the relation between the GDE and the GIE led to the conclusion that the area based GDE does not suffice to explain the improvements of the GIE that were obtained with H^1 modally curved meshes. It is suspected that a measure based on the mesh normal could fill this gap. In the literature, several measures based on differential geometric aspects, such as the normal, were found. These could be applied to the curved meshes and tested for their significance in the modelling of the GIE. However, all these measures are not independent of the curve parametrisation. The development of a parametrisation-independent GDE measure could be another interesting research direction.
For the generalisation of the link between the GDE and the GIE, it would have been preferable to consider a relative area based measure for the circle test case, similar to the measure for the distorted ellipse and the 2D generic intake, and the corresponding volume based measures in 3D. In [99], the reliability of different GDE measures is discussed in general, triggered by a comment of one of the reviewers regarding the example of a distorted circle with radius R of the form $x = R' \cos(\theta)$, $y = R' \sin(\theta)$, with $R' = R + a \cos(n\theta)$. The comparison of geometries with different values of n shows that neither an area based nor a GDE based on the normal on its own is reasonable for all possible geometries. This further enhances the idea to combine these two GDE measures to obtain a reliable overall measure.

8.2.3 Open Questions

An open question that was not deeper examined is the stepwise behaviour for the results of the sphere test case when it is curved with any but the reference based curving method. For the latter, an explanation is found and provided in Section 3.3.1.
Also not further investigated is the unexpected early stagnation of the GDE_V evaluated towards the exact sphere for the unit sphere test case in Section 6.1.5.

Appendix A

2D Blending for Reference-based Modal Curving

In this appendix, the blending function for the reference-based modal curving is derived. Note that the modally curved edges are represented as a linear combination of Lobatto basis polynomials in the reference space, but not in the physical space. For this reason, the Lobatto shape functions cannot be applied directly.
The constraints (4.9) are analogous to the property of the modal edge shape functions $\varphi_{T,k}^{e_i}(\boldsymbol{\xi})$ on the triangle. The shape function is zero on edges e_j, $j \neq i$, because the product of the two barycentric coordinates in (2.6) is zero on those edges. Along edge e_i, the argument $\lambda_{i+1}(\boldsymbol{\xi}) - \lambda_i(\boldsymbol{\xi})$ of the kernel function evolves linearly from -1 to 1 and $l_0(\lambda_{i+1}(\boldsymbol{\xi}) - \lambda_i(\boldsymbol{\xi})) = \lambda_i$ and $l_1(\lambda_{i+1}(\boldsymbol{\xi}) - \lambda_i(\boldsymbol{\xi})) = \lambda_{i+1}$. Therefore, the trace of $\varphi_{T,k}^{e_i}(\boldsymbol{\xi})$ along e_i corresponds to l_k.
Consequently, the function

$$\boldsymbol{\kappa}^{e_i} = \lambda_i \lambda_{i+1} \boldsymbol{d}^{e_i} (\lambda_{i+1} - \lambda_i) \frac{1}{l_0(\lambda_{i+1} - \lambda_i) l_1(\lambda_{i+1} - \lambda_i)}$$

fulfils the constraints (4.9), with e_i the curved edge.
Since $\boldsymbol{\kappa}^{e_i}$ has only influence on edge e_i, this blending approach can also be applied to elements with two or three curved edges with independent edge curving orders.

The function $\boldsymbol{d}$ depends on the formula of the curved edge. For a reference-based modally curved edge e with vertices $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, quadrangle vertices $\boldsymbol{q}_{e,i}$, $i = 1, \ldots, 4$ and Lobatto coefficients α_j, $j = 2, \ldots, q$, the formula is

$$\begin{aligned}
\boldsymbol{f}_c(\tilde{\xi}) &= F(\xi(\tilde{\xi}), \eta(\tilde{\xi})) \\
&= F(\tilde{\xi}, \sum_{i=2}^{q} l_i(\tilde{\xi}) \alpha_i) \\
&= \boldsymbol{c}_{e_i,1} \tilde{\xi} \left(\sum_{i=2}^{q} l_i(\tilde{\xi}) \alpha_i \right) + \boldsymbol{c}_{e_i,2} \tilde{\xi} + \boldsymbol{c}_{e_i,3} \left(\sum_{i=2}^{q} l_i(\tilde{\xi}) \alpha_i \right) + \boldsymbol{c}_{e_i,4},
\end{aligned}$$

with the constants

$$\begin{aligned}
\boldsymbol{c}_{e_i,1} &= \frac{1}{4}\left(\boldsymbol{q}_{e_i,1} - \boldsymbol{q}_{e_i,2} + \boldsymbol{q}_{e_i,3} - \boldsymbol{q}_{e_i,4}\right) \\
\boldsymbol{c}_{e_i,2} &= \frac{1}{4}\left(-\boldsymbol{q}_{e_i,1} + \boldsymbol{q}_{e_i,2} + \boldsymbol{q}_{e_i,3} - \boldsymbol{q}_{e_i,4}\right) \\
\boldsymbol{c}_{e_i,3} &= \frac{1}{4}\left(-\boldsymbol{q}_{e_i,1} - \boldsymbol{q}_{e_i,2} + \boldsymbol{q}_{e_i,3} + \boldsymbol{q}_{e_i,4}\right) \\
\boldsymbol{c}_{e_i,4} &= \frac{1}{4}\left(\boldsymbol{q}_{e_i,1} + \boldsymbol{q}_{e_i,2} + \boldsymbol{q}_{e_i,3} + \boldsymbol{q}_{e_i,4}\right).
\end{aligned}$$

The straight sided edge is parametrised as

$$\boldsymbol{f}_s(\tilde{\xi}) = \boldsymbol{v}_1 + \frac{\tilde{\xi}+1}{2}(\boldsymbol{v}_2 - \boldsymbol{v}_1).$$

Further, the edge vertices can be represented as

$$\begin{aligned}
\boldsymbol{v}_1 &= \boldsymbol{q}_{e_i,1} + \frac{1}{2}(\boldsymbol{q}_{e_i,4} - \boldsymbol{q}_{e_i,1}) \\
\boldsymbol{v}_2 &= \boldsymbol{q}_{e_i,2} + \frac{1}{2}(\boldsymbol{q}_{e_i,3} - \boldsymbol{q}_{e_i,2}),
\end{aligned}$$

which can be used to verify the equation

$$\boldsymbol{c}_{e_i,2}\tilde{\xi} + \boldsymbol{c}_{e_i,4} = \boldsymbol{v}_1 + \frac{\tilde{\xi}+1}{2}(\boldsymbol{v}_2 - \boldsymbol{v}_1).$$

All together, the formula for $\boldsymbol{\kappa}^{e_i}$ can be derived as

$$\begin{aligned}
\boldsymbol{\kappa}^{e_i} &= \lambda_i\lambda_{i+1}\frac{\boldsymbol{d}^{e_i}(\tilde{\xi})}{l_0(\tilde{\xi})l_1(\tilde{\xi})} \\
&= \lambda_i\lambda_{i+1}\left(\boldsymbol{c}_{e_i,1}\tilde{\xi}\sum_{i=2}^{q}\phi_{i-2}(\tilde{\xi})\alpha_i + \boldsymbol{c}_{e_i,3}\sum_{i=2}^{q}\phi_{i-2}(\tilde{\xi})\alpha_i + \frac{\boldsymbol{c}_{e_i,2}\tilde{\xi} + \boldsymbol{c}_{e_i,4} - \boldsymbol{v}_1 - \frac{\tilde{\xi}+1}{2}(\boldsymbol{v}_2 - \boldsymbol{v}_1)}{l_0(\tilde{\xi})l_1(\tilde{\xi})}\right) \\
&= \lambda_i\lambda_{i+1}\left(\boldsymbol{c}_{e_i,1}\tilde{\xi} + \boldsymbol{c}_{e_i,3}\right)\sum_{i=2}^{q}\phi_{i-2}(\tilde{\xi})\alpha_i.
\end{aligned}$$

Appendix B

3D Curving: Details of Edge Curving

As described in Section 6.1.2.1, the detection of the target curve is done in a traversal algorithm. It travels from the start node to the end node by moving along intersection points between the Cmesh normal plane and the Mmesh edges. In the implementation, intersections are associated with the corresponding Mmesh. If an intersection coincides with an Mmesh vertex, it is therefore associated with several Mmesh edges.
Typically, several paths are possible, belonging to different initial edges. All paths are tested until they either reach the end node or a dead end. In the end, the shortest successful path is selected.
For a general traversal step, the next edges with intersections are detected. Thereby, the algorithm differs for the two possible cases: The current intersection point either lies on an Mmesh vertex, or only on an Mmesh edge.

(a) **Current intersection coincides with a vertex**

For the case that the current intersection point falls together with an Mmesh vertex, the subsequent intersection point is found with the following algorithm:

1. Detect the index of the Mmesh vertex *currentVertex* of the current intersection point.
2. Obtain the list of all Mmesh faces *connectedFaces* that are connected to *currentVertex*.
3. Find all edges *possibleEdges* that belong to any of *connectedFaces* and that also contain an intersection point.
4. Remove from *possibleEdges* all edges that were already considered for the current path or that are connected to *currentVertex*. The remaining edges are the subsequent edges $nextEdges$.
5. Mark all edges connected to *currentVertex* as considered for the current path.
6. Return the $nextEdges$ and the updated list of considered edges.

(b) **Current intersection lies only on an edge**

In the more general case, the current intersection point does not coincide with an Mmesh vertex, but lies on one edge that is called *currentEdge*. The algorithm that was described above simplifies to

1. Obtain the Mmesh faces *connectedFaces* that are connected to the *currentEdge*.

2. Find all edges *possibleEdges* that belong to any of *connectedFaces* and that also contain an intersection point.
3. Remove from *possibleEdges* all edges that were already considered for the current path. The remaining edges are the subsequent edges $nextEdges$.
4. Return the $nextEdges$.

The number of edges that were returned as $nextEdges$ decides on the further progression of the algorithm. Based on the algorithms described above, a maximum of 2 edges can be obtained.

(a) **0 edges**

A dead end is reached. The traversal algorithm breaks and the next path is tested.

(b) **1 edge**

The next intersection point lies only on an edge. It is marked as such, the path data is updated and the next traversal step is started from the new intersection point.

(c) **2 edges**

The next intersection point falls together with a vertex. It is checked if this vertex is also the end point.

(A) **The end point is reached.**

One of the $nextEdges$ is added to the path. The path data is updated and the path marked as successful. If another, shorter path was already found before, it is neglected. Otherwise, the path is stored as the currently shortest path and the next path is tested.

(B) **The vertex is not the end point.**

The new intersection point is marked as vertex, the path data is updated with both edges and the next traversal step is started.

Special Treatment for the Starting Edges

At the beginning of the traversal algorithm, the ”next” intersection point is in general not uniquely defined, as the paths can run in both directions from the start point and none of the edges is already marked as ”considered”. Therefore, this case has to be treated separately by the following algorithm:

1. Find all $nextEdges$ starting from the vertex that corresponds to the start point, using the algorithm that was detailed above. They are saved as $secondEdges$ and marked as $notTested$.
2. While not all $secondEdges$ are marked $tested$, try to construct a path with the first edge in $secondEdges$ that is marked $notTested$:
   i. The current $secondEdge$ are marked as $tested$.

ii. It is tested if the other vertex of the current $secondEdge$ corresponds to the end point. If so, the shortest path with two edges is found. The Mmesh is too coarse at this point and the Cmesh edge will not be curved.

iii. All $secondEdges$ that are marked $notTested$ and that share a vertex with the current $secondEdge$ are detected and called $adjacentEdges$. The number of $adjacentEdges$ determines the subsequent steps:

(a) **0** $adjacentEdges$
The intersection point of the current $secondEdge$ lies on the edge (not on any Mmesh vertex). This case is treated as described above.

(b) **1 or more** $adjacentEdges$
A direct test is run to see if the intersection point coincides with one of the two Mmesh vertices that define the current $secondEdge$. A tolerance of $10^{-4} \times$ minimal length of Mmesh edges is applied.

(A) If the intersection point falls together with an Mmesh vertex, an additional test is run to find if there is an $adjacentEdge$ with the same intersection point. Thereby, the same tolerance is applied as for the previous test. If such an $adjacentEdge$ is found, both edges, the current $secondEdge$ and the selected $adjacentEdge$ are then marked as $tested$. Note that the $adjacentEdge$ is in the list of $secondEdges$ as well. Otherwise, only the current $secondEdge$ is added to the path and marked as $tested$. The case is further treated as an intersection on an Mmesh vertex as described above.

(B) If the intersection point does not lie on an Mmesh vertex, it is treated as described above for intersecion points that fall onto an edge.

Appendix C

3D Curving: Details of Face Triangulation

In this appendix, the different cases are listed that were treated in the face triangulation for the 3D face curving (see Section 6.1.3.2).
It still holds that the Cmesh vertices form a subset of the Mmesh vertices. Therefore, the Cmesh vertices can not fall into the interior or an edge of an Mmesh element. This eliminates several cases of intersections, as e.g. the case shown in figure C.1. Further, as the underlying Mmesh is assumed to be well defined, cases with colinear Mmesh vertices won't be considered.

C.1 All-in, 2-in-1-boundary, 1-in-2-boundary, and All-boundary Faces

The four cases All-in, 2-in-1-boundary, 1-in-2-boundary, and All-boundary faces have no intersection with a Cmesh edge normal plane. The original Mmesh triangles are not truncated and can directly be adopted for the triangulation.

C.2 2-in-1-out Faces

For triangles that have two internal and one external node, only one possible case exists (see Fig. C.2). The intersection of the Mmesh element and the internal domain forms a quadrangle. It is triangulated into two triangles by selecting one intersection point and detecting the corresponding internal node on the opposite intersecting Mmesh edge. The selection of the intersection point is

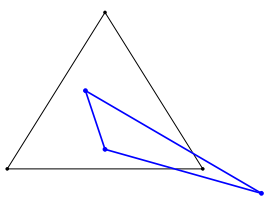


Figure C.1: Case that can not happen, as the Cmesh vertices form a subset of the Mmesh vertices.

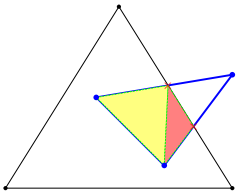


Figure C.2: Cases for 2-in-1-out faces.

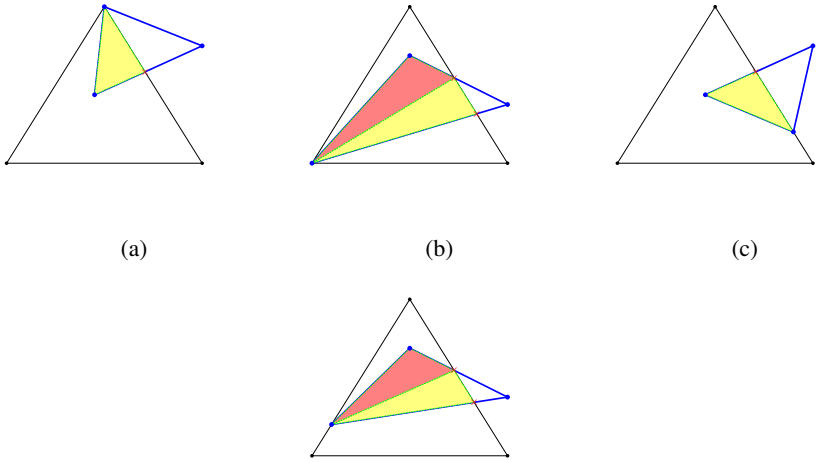


Figure C.3: Cases for 1-in-1-boundary-1-out faces.

done arbitrarily, based on the numbering in the implementation. An additional test could be run to detect the alternative triangulation and to chose the one with the better shaped triangles.

C.3 1-in-1-boundary-1-out Faces

When considering triangles with one vertex on the boundary of the Cmesh element domain, this vertex can either lie on the edge or on an vertex of the domain (see Figure C.3). In the implementation, these two cases are considered separately, but treated the same way. Below, the case with the vertex on the domain edge is described.
If a triangle is defined by one interior node, one exterior node and one node on the boundary, two possible cases can occur, with either one or two intersection points (see Figures C.3(c), (d)). In the case of one intersection point, the intersection of the Mmesh triangle and the domain form a triangle that is defined by replacing the outer vertex of the Mmesh triangle with the intersection point. With two intersection points, the intersection is a quadrangle. It is triangulated into two triangles, following the algorithm described for 2-in-1-out faces in Section C.2.

C.4 2-boundary-1-out Faces

For 2-boundary-1-out faces, both, one, or none of the Mmesh nodes on the boundary can fall together with an Cmesh vertex (see Fig. C.4). Again, these cases are treated separately in the

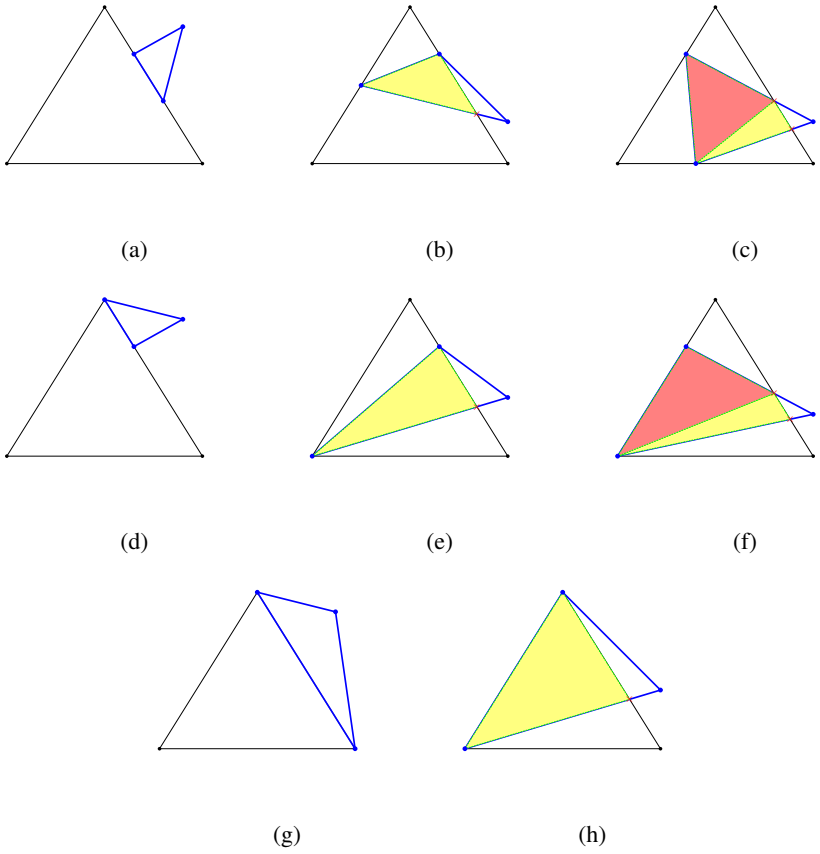


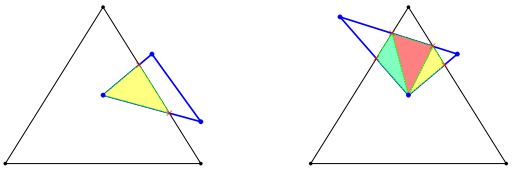
Figure C.4: Cases for 2-boundary-1-out faces.

implementation, but can be pooled for their description.

The treatment depends on the number of intersection points of the Mmesh edges with the domain boundary. If there is no intersection point, the Mmesh lies outside of the domain (see Fig. C.4(a), (d), (g)) and is therefore neglected. With one intersection point, the nodes on the boundary (or Cmesh vertices) and the intersection point form the domain-interior triangle (see Fig. C.4(b), (e), (h)). Two intersection points can occur if only one or none of the boundary nodes lies on a Cmesh vertex (see Fig. C.4(c), (f)). In this case, the intersection with the domain is a quadrangle that is triangulated into two triangles analogously to the algorithm for 2-in-1-out faces in Section C.2.

C.5 1-in-2-out Faces

Mmesh triangles with one interior and two exterior nodes have either two or four intersection points with the domain boundary (see Fig. C.5). In the case of two intersection points, the two exterior nodes are replaced by the intersection points to form the interior triangle. With four intersection points, the intersection of the Mmesh triangle and the domain forms a pentagon that has to be triangulated into three triangles. They are constructed by combining the interior node with the two intersection points with either one of the domain boundary edges, or the two intersection points of the Mmesh edge opposite of the interior edge (see Fig. C.5(b)).



(a) (b)

Figure C.5: Cases for 1-in-2-out faces.

C.6 1-boundary-2-out Faces

As in previous cases, the boundary node can coincide with a Cmesh vertex, or fall onto an edge of the domain. Again, these two cases are treated individually in the implementation but follow the same algorithms and are not described separately here. Mmesh triangles with 1 boundary node and 2 exterior nodes can have no, two, three, or four intersections with the domain boundary. The latter does not occur when the boundary node lies on a Cmesh vertex (see Fig. C.6).
The Mmesh lies completely outside of the domain when no intersection point is detected. It is therefore neglected. With two intersection points, the internal triangle is described by the boundary node and these two intersections. Triangulations are necessary for Mmesh triangles with three or four intersections. There, the intersections form a quadrangle or a pentagon and are triangulated into two or three triangles, respectively. For triangulate the quadrangle, the boundary node is with two intersections that lie together on either an Mmesh edge or an edge of the domain boundary. The triangulation of the pentagon follows the algorithm described in Section C.5. Here, the boundary node replaces the internal node as basic node that is included in all three triangles.

C.7 All-out Faces

Even if all Mmesh nodes lie outside of the domain, there are two possible cases with non-empty intersections. They are classified by either four or six intersection points with the domain boundary (see Fig. C.7).
In all tests for this thesis, the latter never occured Anyway, it would be associated with a very coarse Mmesh. It was therefore neglected in the implementation, returning an error that the Mmesh has to be refined. The intersection would form a hexagon and a triangulation would need four triangles.
If four intersection points are detected, the intersection is a quadrangle that is split into two triangles. The choice of the diagonal edge is arbitrarily chosen, based on the numbering of the intersection points, similar to Section C.2.

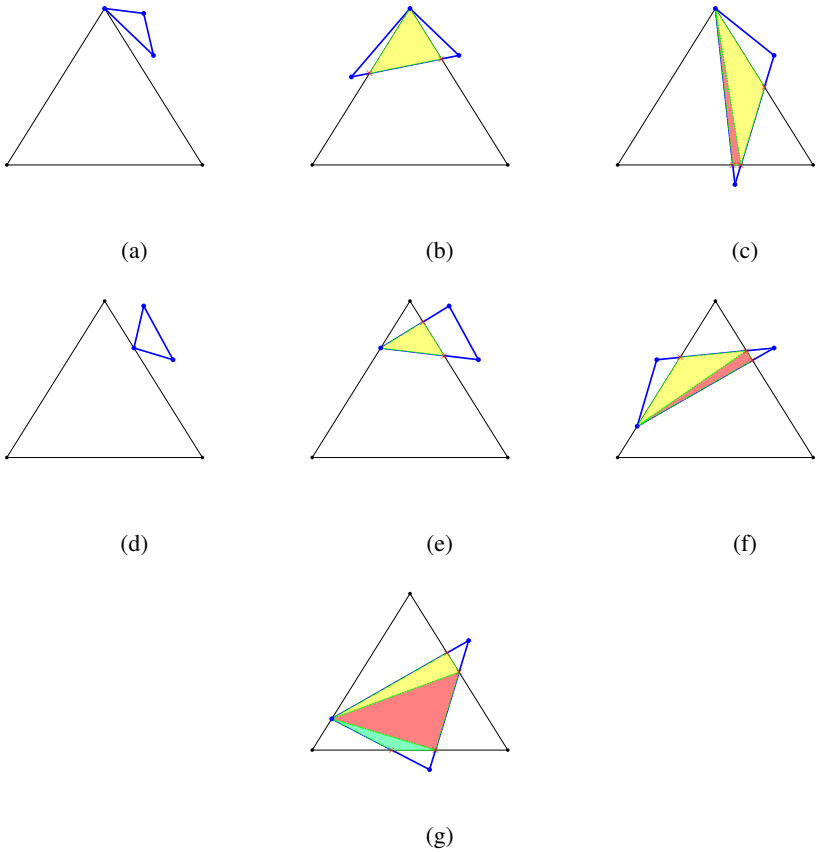


Figure C.6: Cases for 1-boundary-2-out faces.

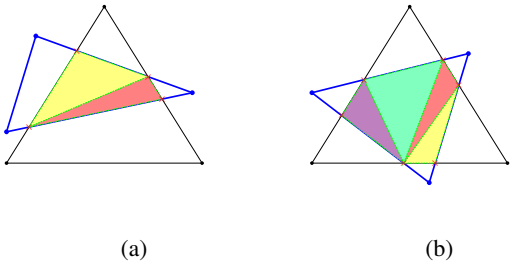


Figure C.7: Cases for all-out faces.

Bibliography

[1] R. Abgrall, C. Dobrzynski, and A. Froehly. A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems. *International Journal for Numerical Methods in Fluids*, 76(4):246–266, 2014.

[2] M. Aliabadi and W. Hall. Nonisoparametric formulations for the three-dimensional boundary element method. *Engineering Analysis*, 5(4):198–204, 1988.

[3] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 13(3-4):251–265, 1995.

[4] ansol - Advanced Numerical Solutions LLC. Plane Wave Scattering by a Rigid Sphere. Website. http://ansol.us/Products/Coustyx/Validation/MultiDomain/Scattering/PlaneWave/HardSphere/Downloads/dataset_description.pdf (version: 2017-03-02).

[5] I. Babuška and B. Guo. Approximation properties of the hp version of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 133(3-4):319–346, 1996.

[6] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics*, 138(2):251 – 285, 1997.

[7] H. Bériot, G. Gabard, and E. Perrey-Debain. Analysis of high-order finite elements for convected wave propagation. *International Journal for Numerical Methods in Engineering*, 96(11):665–688, 2013.

[8] H. Bériot, E. Perrey-Debain, M. B. Tahar, and C. Vayssade. Plane wave basis in Galerkin BEM for bidimensional wave scattering. *Engineering Analysis with Boundary Elements*, 34(2):130–143, 2010.

[9] H. Bériot, A. Prinn, and G. Gabard. Efficient implementation of high-order finite elements for Helmholtz problems. *International Journal for Numerical Methods in Engineering*, 106(3):213–240, 2016. nme.5172.

[10] H. Bériot and M. Tournour. On the locally-conformal perfectly matched layer implementation for helmholtz equation. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 2009, pages 503–513. Institute of Noise Control Engineering, 2009.

[11] A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. *Journal of Computational Physics*, 223(2):469 – 488, 2007.

[12] K. Bock and J. Stiller. Energy-minimizing curve fitting for high-order surface mesh generation. *Applied Mathematics*, 5(21):3318, 2014.

[13] K. Bock and J. Stiller. Generation of high-order polynomial patches from scattered data. In *Spectral and High Order Methods for Partial Differential Equations-ICOSAHOM 2012*, pages 157–167. Springer, 2014.

[14] K. Bock and J. Stiller. Optimizing triangular high-order surface meshes by energy-minimization. *Engineering with Computers*, 34(4):659–670, 2018.

[15] A. Cohen, M. A. Davenport, and D. Leviatan. On the stability and accuracy of least squares approximations. *Foundations of Computational Mathematics*, 13(5):819–834, Oct 2013.

[16] G. Dahlquist and A. Björck. *Numerical methods*. Prentice-Hall, 1974.

[17] J. Danczyk and K. Suresh. Finite element analysis over tangled simplicial meshes: Theory and implementation. *Finite Elements in Analysis and Design*, 70:57–67, 2013.

[18] L. Demkowicz, P. Gatto, W. Qiu, and A. Joplin. G^1-interpolation and geometry reconstruction for higher order finite elements. *Computer Methods in Applied Mechanics and Engineering*, 198(13):1198–1212, 2009.

[19] L. Demkowicz, W. Rachowicz, and P. Devloo. A fully automatic hp-adaptivity. *Journal of Scientific Computing*, 17(1-4):117–142, 2002.

[20] A. Deraemaeker, I. Babuška, and P. Bouillard. Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions. *International journal for numerical methods in engineering*, 46(4):471–499, 1999.

[21] W. Desmet and P. Sas. Introduction to numerical acoustics. In *ISAAC25, Applied & Numerical Acoustics*, 2014.

[22] S. Dey, M. S. Shephard, and J. E. Flaherty. Geometry representation issues associated with p-version finite element computations. *Computer Methods in Applied Mechanics and Engineering*, 150(1):39–55, 1997.

[23] J. Escobar, R. Montenegro, G. Montero, E. Rodríguez, and J. González-Yuste. Smoothing and local refinement techniques for improving tetrahedral mesh quality. *Computers & structures*, 83(28):2423–2430, 2005.

[24] J. Escobar, E. Rodrıguez, R. Montenegro, G. Montero, and J. González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2775–2787, 2003.

[25] European Commission. Flightpath 2050, Europe's vision for aviation, 2011.

[26] J. H. Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.

[27] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(3):231–250, 1997.

[28] M. S. Floater. Mean value coordinates. *Computer aided geometric design*, 20(1):19–27, 2003.

[29] L. A. Freitag and P. M. Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *International Journal for Numerical Methods in Engineering*, 53(6):1377–1391, 2002.

[30] G. Gabard and R. Astley. Theoretical model for sound radiation from annular jet pipes: far-and near-field solutions. *Journal of Fluid Mechanics*, 549:315–341, 2006.

[31] G. Gabard, H. Bériot, A. Prinn, and K. Kucukcoskun. Adaptive, high-order finite-element method for convected acoustics. *AIAA Journal*, 56(8):3179–3191, 2018.

[32] A. Gargallo-Peiró, X. Roca, J. Peraire, and J. Sarrate. Defining quality measures for validation and generation of high-order tetrahedral meshes. In J. Sarrate and M. Staten, editors, *Proceedings of the 22nd International Meshing Roundtable*, pages 109–126. Springer International Publishing, 2014.

[33] A. Gargallo-Peiró, X. Roca, J. Peraire, and J. Sarrate. Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 2015.

[34] A. Gargallo-Peiró, X. Roca, and J. Sarrate. A surface mesh smoothing and untangling method independent of the CAD parameterization. *Computational Mechanics*, 53(4):587–609, 2014.

[35] C. Geuzaine, A. Johnen, J. Lambrechts, J.-F. Remacle, and T. Toulorge. The generation of valid curvilinear meshes. In *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, pages 15–39. Springer, 2015.

[36] 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, 2009.

[37] M. Guthe, P. Borodin, and R. Klein. Fast and accurate Hausdorff distance calculation between meshes. 2005.

[38] K. Hamiche, G. Gabard, and H. Bériot. A high-order finite element method for the linearised Euler equations. *Acta Acustica united with Acustica*, 102(5):813–823, 2016.

[39] J. Hesthaven and T. Warburton. *High-order/spectral Methods on Unstructured Grids*. NASA Langley Research Center. Institute for Computer Applications in Science , 2001.

[40] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[41] S. Hygge, G. W. Evans, and M. Bullinger. A prospective study of some effects of aircraft noise on cognitive performance in schoolchildren. *Psychological Science*, 13(5):469–474, 2002.

[42] F. Ihlenburg and I. Babuška. Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation. *International journal for numerical methods in engineering*, 38(22):3745–3774, 1995.

[43] F. Ihlenburg and I. Babuška. Finite element solution of the Helmholtz equation with high wave number Part I: The h-version of the FEM. *Computers & Mathematics with Applications*, 30(9):9 – 37, 1995.

[44] F. Ihlenburg and I. Babuska. Finite element solution of the Helmholtz equation with high wave number part II: the hp version of the FEM. *SIAM Journal on Numerical Analysis*, 34(1):315–358, 1997.

[45] J. Ims, Z. Duan, and Z. Wang. meshCurve: An automated low-order to high-order mesh generator. In *AIAA 22nd Aeroacoustics Conference*, Dallas, Texas, USA, 2015. AIAA Paper 2015-2293.

[46] J. Ims and Z. Wang. Automated low-order to high-order mesh conversion. *Engineering with Computers*, pages 1–13, 2018.

[47] X. Jiao and N. R. Bayyana. Identification of C^1 and C^2 discontinuities for surface meshes in CAD. *Computer-Aided Design*, 40(2):160–175, 2008.

[48] X. Jiao and D. Wang. Reconstructing high-order surfaces for meshing. *Engineering with Computers*, 28(4):361–373, 2012.

[49] X. Jiao, D. Wang, and H. Zha. Simple and effective variational optimization of surface and volume triangulations. *Engineering with Computers*, 27(1):81–94, 2011.

[50] G. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.

[51] K. Lipnikov and Y. Vassilevski. On discrete boundaries and solution accuracy in anisotropic adaptive meshing. *Engineering with Computers*, 26(3):281–288, 2010.

[52] X. Luo. *An automatic adaptive directional variable p-version method in 3D curved domains*. PhD thesis, Rensselaer Polytechnic Institute, 2005.

[53] X. Luo, M. Shephard, L.-Q. Lee, C. Ng, and L. Ge. Tracking adaptive moving mesh refinements in 3D curved domains for large-scale higher order finite element simulations. In R. Garimella, editor, *Proceedings of the 17th International Meshing Roundtable*, pages 585–601. Springer Berlin Heidelberg, 2008.

[54] X. Luo, M. S. Shephard, and J.-F. Remacle. The influence of geometric approximation on the accuracy of high order methods. *Rensselaer SCOREC report*, 1, 2001.

[55] X. Luo, M. S. Shephard, J.-F. Remacle, R. M. O'Bara, M. W. Beall, B. Szabó, and R. Actis. p-version mesh generation issues. In *Proceedings of the 11th International Meshing Roundtable*, pages 343–354, 2002.

[56] A. Mahashabde, P. Wolfe, A. Ashok, C. Dorbian, Q. He, A. Fan, S. Lukachko, A. Mozdzanowska, C. Wollersheim, S. R. Barrett, M. Lockec, and I. A. Waitz. Assessing the environmental impacts of aircraft noise and emissions. *Progress in Aerospace Sciences*, 47(1):15–52, 2011.

[57] A. M. McIvor and R. J. Valkenburg. A comparison of local surface geometry estimation methods. *Machine Vision and Applications*, 10(1):17–26, 1997.

[58] P. Morrell and C. H.-Y. Lu. Aircraft noise social cost and charge mechanisms–a case study of Amsterdam Airport Schiphol. *Transportation Research Part D: Transport and Environment*, 5(4):305–320, 2000.

[59] A. Mosig and M. Clausen. Approximately matching polygonal curves with respect to the Fréchet distance. *Computational Geometry*, 30(2):113–127, 2005.

[60] D. Moxey, D. Ekelschot, Ü. Keskin, S. Sherwin, and J. Peiró. A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality. *Procedia Engineering*, 82:127–135, 2014.

[61] D. Moxey, M. Green, S. 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.

[62] R. Munt. The interaction of sound with a subsonic jet issuing from a semi-infinite cylindrical pipe. *Journal of Fluid Mechanics*, 83(4):609–640, 1977.

[63] NS (https://de.mathworks.com/matlabcentral/profile/authors/1640586-ns). Curve intersections. MathWorks File Exchange. URL:https://de.mathworks.com/matlabcentral/fileexchange/22441-curve-intersections (Version: 3.0, 21 Sept. 2010).

[64] A. Ooms (https://math.stackexchange.com/users/63427/aurélien ooms). Find if three points in 3-dimensional space are collinear. Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/1778739 (version: 2016-05-09).

[65] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proceedings of the conference on Visualization'02*, pages 163–170. IEEE Computer Society, 2002.

[66] G. M. Phillips. *Interpolation and approximation by polynomials*, volume 14. Springer Science & Business Media, 2003.

[67] R. Plato. Numerische Mathematik kompakt. *Grundlagenwissen für Studium und Praxis. Vieweg Wiesbaden*, 2006.

[68] Pointwise, Inc. High-Order Mesh Generation at Pointwise. Website. http://www.pointwise.com/theconnector/2016-Q3/High-Order-Mesh-Generation-at-Pointwise.html (version: 2019-11-14).

[69] R Core Team. *R / A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[70] Z. Rarata. Application and assessment of time-domain DGM for intake acoustics using 3D linearized Euler equations. August 2014.

[71] N. Ray, T. Delaney, D. R. Einstein, and X. Jiao. Surface remeshing with robust high-order reconstruction. *Engineering with Computers*, 30(4):487–502, 2014.

[72] J.-F. Remacle, J. Lambrechts, C. Geuzaine, and T. Toulorge. Optimizing the geometrical accuracy of 2D curvilinear meshes. *Procedia Engineering*, 82:228–239, 2014.

[73] S. W. Rienstra and A. Hirschberg. An introduction to acoustics. *Eindhoven University of Technology*, 18:19, 2003.

[74] M. Rosenlund, N. Berglind, G. Pershagen, L. Järup, and G. Bluhm. Increased prevalence of hypertension in a population exposed to aircraft noise. *Occupational and Environmental Medicine*, 58(12):769–773, 2001.

[75] E. Ruiz-Gironés, J. Sarrate, and X. Roca. Defining an $\mathcal{L}^2$-disparity measure to check and improve the geometric accuracy of non-interpolating curved high-order meshes. *Procedia Engineering*, 124:122–134, 2015.

[76] D. P. Sanjaya and K. J. Fidkowski. Improving high-order finite element approximation through geometrical warping. In *AIAA 22nd Aeroacoustics Conference*, Dallas, Texas, USA, 2015. AIAA Paper 2015-2605.

[77] V. S. Schmid, H. Bériot, O. Atak, and G. Gabard. High-order curved mesh generation by using a fine linear target mesh. In *ECCOMAS Congress 2016 Proceedings*, volume 1, pages 493–503. ECCOMAS, 2016.

[78] R. Sevilla and S. Fernández-Méndez. Numerical integration over 2D NURBS-shaped domains with applications to NURBS-enhanced FEM. *Finite Elements in Analysis and Design*, 47(10):1209 – 1220, 2011.

[79] R. Sevilla, S. Fernández-Méndez, and A. Huerta. NURBS-enhanced finite element method (NEFEM). *0029-5981*, 76(1):56–83, 2008.

[80] R. Sevilla, S. Fernández-Méndez, and A. Huerta. 3D NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 88(2):103–125, 2011.

[81] R. Sevilla, S. Fernández-Méndez, and A. Huerta. NURBS-Enhanced Finite Element Method (NEFEM) a seamless bridge between CAD and FEM. *Archives of Computational Methods in Engineering*, 18(4):441–484, 2011.

[82] P. Šolín, K. Segeth, and I. Doležel. Higher order finite element methods, 2004.

[83] P. Šolín, T. Vejchodskỳ, M. Zítka, and F. Ávila. Imposing orthogonality to hierarchic higher-order finite elements. *Mathematics and Computers in Simulation*, 76(1):211–217, 2007.

[84] A. Sommerfeld. Die Greensche Funktion der Schwingungsgleichung. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 21:309–353, 1912.

[85] M. A. Taylor, B. A. Wingate, and R. E. Vincent. An algorithm for computing Fekete points in the triangle. *SIAM Journal on Numerical Analysis*, 38(5):1707–1720, Oct. 2000.

[86] R. L. Taylor. On completeness of shape functions for finite element analysis. *International Journal for Numerical Methods in Engineering*, 4(1):17–22, 1972.

[87] K. W. Thompson. Time dependent boundary conditions for hyperbolic systems. *Journal of Computational Physics*, 68(1):1–24, 1987.

[88] L. L. Thompson. A review of finite-element methods for time-harmonic acoustics. *Journal of the Acoustical Society of America*, 119(3):1315–1330, 2006.

[89] T. Toulorge and W. Desmet. Curved boundary treatments for the discontinuous Galerkin method applied to aeroacoustic propagation. *AIAA journal*, 48(2):479–489, 2010.

[90] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts. Robust untangling of curvilinear meshes. *Journal of Computational Physics*, 254(0):8 – 26, 2013.

[91] Y. V. Vassilevski, V. Dyadechko, and K. Lipnikov. Hessian-based anisotropic mesh adaptation in domains with discrete boundaries. *Russian Journal of Numerical Analysis and Mathematical Modelling rnam*, 20(4):391–402, 2005.

[92] P. Volino and N. M. Thalmann. The spherigon: a simple polygon patch for smoothing quickly your polygonal meshes. In *Computer Animation 98. Proceedings*, pages 72–78. IEEE, 1998.

[93] Z. Wang, Y. Li, F. Jia, G. Laskowski, J. Kopriva, U. Paliath, and R. Bhaskaran. Towards industrial large eddy simulation using the fr/cpr method. *Computers & Fluids*, 156(Supplement C):579 – 589, 2017. Ninth International Conference on Computational Fluid Dynamics (ICCFD9).

[94] C. Wenk, R. Salas, and D. Pfoser. Addressing the need for map-matching speed: Localizing global curve-matching algorithms. In *Scientific and Statistical Database Management, 2006. 18th International Conference on*, pages 379–388. IEEE, 2006.

[95] World Health Organization. Burden of disease from environmental noise, 2011.

[96] Z. Xie, R. Sevilla, O. Hassan, and K. Morgan. The generation of arbitrary order curved meshes for 3D finite element analysis. *Computational Mechanics*, 51(3):361–374, 2013.

[97] D. Xue and L. Demkowicz. Control of geometry induced error in hp finite element (FE) simulations. I. Evaluation of FE error for curvilinear geometries. *International Journal of Numerical Analysis and Modeling*, 2(3):283–300, 2005.

[98] V. S. Ziel, H. Bériot, O. Atak, and G. Gabard. Comparison of 2D boundary curving methods with modal shape functions and a piecewise linear target mesh. *Procedia Engineering*, 203, 2017.

[99] V. S. Ziel, H. Bériot, O. Atak, and G. Gabard. High-order 2D mesh curving methods with a piecewise linear target and application to Helmholtz problems. *Computer-Aided Design*, 105:26–41, 2018.

[100] O. Zienkiewicz and R. Taylor. The finite element method, volume 1: The basis, 2000.