

Body-fitting Meshes for the Discontinuous Galerkin Method

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt

zur Erlangung
der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

DISSERTATION

von
Jian Cui, M.Sc.
geboren am 04. August 1982 in Liaoning, China

Referent: Prof. Dr.-Ing. Thomas Weiland
Korreferent: Prof. Dr.-Ing. Min Zhang

Tag der Einreichung: 05.02.2013
Tag der mündlichen Prüfung: 17.06.2013

D 17
Darmstädter Dissertation

Darmstadt 2013

Randangepasste Gitter für die Discontinuous Galerkin Methode

Kurzfassung der Arbeit

In der vorliegenden Dissertation wird ein Verfahren zur Erzeugung hochpräziser, randangepasster Gitter vorgestellt und im Rahmen der Diskontinuierlichen Galerkin (DG) Methode angewendet. Im Gegensatz zu den meisten anderen Gittergeneratoren erzeugt das vorgestellte Verfahren ein Gitter, welches ausschließlich aus vierseitigen oder hexaedrischen Elementen besteht. Die hohe Qualität der Randapproximation wird mittels gekrümmter Elemente hoher Ordnung erreicht. Die erzeugten Gitter sind bestens für den Einsatz von numerischen Methoden höherer Ordnung geeignet. Dies gilt insbesondere für die DG Methode, da große Teile des Gitters vollständig strukturiert sind und zugleich die Ordnung der Geometrierepräsentation der hohen Ordnung der numerischen Methode entsprechen kann.

Die Gittererzeugung erfolgt in zwei Schritten. Zunächst wird ein inneres Volumen ausgewählt, welches mittels eines vollständig strukturierten kartesischen Gitters abgebildet wird. Im Anschluss werden alle eingebetteten Objekte sowie die Randflächen mittels einer Pufferschicht mit dem inneren Gitter verbunden. Innerhalb dieser Schicht werden gekrümmte Elemente hoher Ordnung verwendet, wodurch eine hohe Approximationsgenauigkeit nicht ebener Objekte und Ränder erzielt wird. Die resultierenden Gitter sind optimal für die Anwendung numerischer Methoden hoher Ordnung geeignet, da einerseits eine hohe Effizienz durch die kartesische Struktur im Inneren und andererseits eine hohe Genauigkeit, selbst in Anwesenheit gekrümmten Objekte und Ränder, gegeben ist.

Nach der Beschreibung des Gittergenerierungsalgorithmus, wird die Diskontinuierliche Galerkin Methode für die Maxwell'schen Gleichungen beschrieben. Die DG Methode erlaubt die Verwendung von Basisfunktionen beliebig hoher Ordnung. Damit kann für hinreichend glatte Lösungen exponentielle Konvergenz des Fehlers bezüglich der Anzahl der Freiheitsgrade erreicht werden. Es wird anhand von Beispielen gezeigt, dass mit der DG Methode und dem in dieser Arbeit vorgestellten Gittergenerierungsverfahren exponentielle Konvergenz für nicht-geradlinig berandete Rechengebiete erreicht werden kann. Weiterhin wird exemplarisch für einen Resonator gezeigt, dass die mittels Fouriertransformation eines Zeitsignals erhaltenen Spektren frei von nicht-physikalischen Eigenwerten sind.

Body-fitting meshes for the Discontinuous Galerkin Method

Abstract

In this dissertation, a scheme capable of generating highly accurate, body fitting meshes and its application with the Discontinuous Galerkin Method is introduced. Unlike most other mesh generators, this scheme generates meshes consisting of quadrilateral or hexahedral elements exclusively. The high approximation quality is achieved by means of high order curved elements. The resulting meshes are highly suited for the application with high order methods, specifically the Discontinuous Galerkin Method, as large portions of the mesh are fully structured while matching the high order of the numerical method with a high order geometry representation at object and domain boundaries.

The mesh scheme works in two steps. It chooses an interior volume that can be represented using a fully structured Cartesian mesh and connects it to the embedded objects and domain boundaries with a so called buffer-layer in a second step. Inside the layer, high order curved elements are applied for yielding high geometric representation accuracy. The resulting meshes are ideal for the application of high order methods. In the interior part the Cartesian structure can be exploited for obtaining high efficiency of the numerical method while the accuracy potential can be realized also in the presence of curved objects and boundaries.

After introducing the mesh scheme, the Discontinuous Galerkin Method is described and applied for solving Maxwell's equations. As a high order method it achieves exponential convergence under p-refinement. It is shown that using meshes produced by the novel scheme this property is achieved for curved domains as well. As an example, optimal convergence rates are demonstrated in a cylindrical cavity problem. In another example, the abilities of the method to produce correct spectral properties of closed resonator problems are investigated. To this end, a time-domain signal is recorded during the transient analysis. After applying the Fourier transform accurate frequency spectra are observed, which are free of spurious modes.

Contents

1. Introduction	1
1.1. Motivation and Objectives	1
1.1.1 High Order Finite Element Method	2
1.1.2 High Order Geometric Description.....	5
1.2. Overview.....	7
2. Body-fitting Mesh Scheme	9
2.1. Mesh Generation for Curved Geometries	11
2.1.1 Background Mesh Generation	11
2.1.2 Mesh Schemes for Curved Boundaries.....	13
2.1.3 All Quadrilateral (Hexahedral) Meshes	16
2.2. Implicit and Explicit Form of Boundaries	20
2.2.1 Curved Boundaries with the Buffer Layer Scheme	23
2.2.2 Applications with Buffer Elements.....	26
2.3. High Order Scheme with Curved Elements.....	29
2.3.1 Curved Elements with the Transfinite Interpolation Technique.....	30

2.3.2	Numerical Integration	34
2.4.	Extending the Mesh Scheme into Three Dimensions	36
3.	Discontinuous Galerkin Method	41
3.1.	High Order Finite Element Methods.....	41
3.2.	The Discontinuous Galerkin Method and its Properties	46
3.3.	Maxwell's Equations	47
3.4.	Discretization of the Maxwell's Equations.....	49
3.4.1	Governing Equations	49
3.4.2	Semidiscrete Formulation	50
4.	Eigenvalue Problems in Electromagnetics	57
4.1.	High Order Accuracy in a Square Domain	58
4.2.	Electromagnetic Fields in a Cylindrical Domain.....	61
4.2.1	Electromagnetic Solutions in Cylindrical Coordinates.....	61
4.2.2	Wave Propagation in a Circular Domain	63
4.2.3	High Order Convergence	65
4.3.	Frequency Spectrum Analysis	70
5.	Summary and Outlook.....	75
5.1.	Summary.....	75
5.2.	Future Work.....	76
Appendix A	79

List of Figure	81
Bibliography	85
Acknowledgements	91
Curriculum Vitae	93

1. Introduction

In recent years high order methods have been studied and applied intensively in variety of fields, such as electromagnetics or fluid dynamics. While high order methods such as, e.g., the Discontinuous Galerkin Method (DGM or DG method) are known for forty years [1] mainly low order methods such as Finite Difference Methods [2] were used because they are less demanding in terms of computational resources. The steady increase of computing power has shifted the focus back to high order methods, which are known to yield higher accuracy per Degree of Freedom (DoF) [3].

1.1. Motivation and Objectives

A remarkable property of the DGM is that exponential convergence can be obtained, i.e., the best convergence behavior possible [4]. In this case, the approximation error decreases exponentially with the number of DoF. This can only be achieved with methods, which allow for increasing the approximation order, i.e. not with low order Finite Difference Methods. However, practically this is often limited by the approximation of the domain of interest using a polygonal or polyhedral shape. In this case the boundaries (surfaces) of the domain are piece-wise linear. Elements with straight edges are often not capable of covering the domain exactly, leading to a rough truncation of curved features and domains.

In order to maintain exponential convergence in the presence of curved objects, the boundaries (surfaces) have to be described with high accuracy as well. The study in [5] shows that meaningful high order accurate results can be obtained only if the curved boundaries are represented with high order geometric approximations. For this purpose, the elements of the computational mesh are required to be curved such that they conform to the domain of interests and its embedded objects. This can be achieved by pushing the edges and faces of the mesh elements onto the exact boundary as proposed in [6].

While most implementations such as [1, 3, 5, 6] employ meshes composed of triangular or tetrahedral elements, we propose an alternative mesh scheme based on Cartesian grids. It generates quadrilateral or hexahedral meshes in a simple process for both, exact

geometries and objects represented by Non-Uniform Rational B-Splines (NURBS) [7]. The resulting meshes will be exploited in numerical implementations for the electromagnetics problems. The advantage of quadrilateral and hexahedral meshes is that tensor product bases can be applied, which will be shown in Chapter 3 to have a number of advantages. Validations are carried out using analytical results.

The mesh scheme offers many advantages due to its comparatively simple process of mesh generation which can be summarized as follows: A structured so-called background mesh is outlined, where the node placements are identified by global indices of the coordinates. This allows for identifying elements by index arithmetic rather than through explicit storage of element connectivity in memory. The boundary fitting process is controlled by the boundary layer. Therefore, the mesh is unstructured only near curved boundaries instead of all over the domain. In addition, well-shaped elements are guaranteed without performing the procedure of inserting and deleting nodes to improve mesh quality.

1.1.1 High Order Finite Element Method

The Discontinuous Galerkin Method (DGM) [8, 9, 10, 11] is a high order finite element method. A complete introduction of the DGM will be given in Chapter 3. For readers who have experience with the DGM, may skip the one dimensional example of this section and focus on the curved mesh refinement in Section 2.1.2.

In order to present the advantages of a high order method, an electromagnetic problem is computed with the DGM in one spatial dimension. It illustrates the typical finite element analysis steps.

First, the physical problem needs to be modeled. In one dimension, the Maxwell's equations can be written as scalar wave equations. Therefore, the modeling of the electromagnetic waves is equivalent to a one-dimensional boundary value problem. The system of equations read

$$\varepsilon \frac{\partial E}{\partial t} + \frac{\partial H}{\partial x} = 0, \quad (1-1)$$

$$\mu \frac{\partial H}{\partial t} + \frac{\partial E}{\partial x} = 0 \quad \text{in } \Omega, \quad (1-2)$$

$$E(a, t) = E(b, t) = 0 \quad \text{on } \Gamma, \quad (1-3)$$

with the scalar electric field E , magnetic field H , electric permittivity ε and magnetic permeability μ . The domain of interest Ω is the interval $[a, b] = [-1, 1]$. In the one-dimensional case, the boundary Γ consists of the ending points a and b .

A discretization is simply obtained by partitioning the domain Ω into N disjoint intervals $\cup \Omega_j = \Omega$, where $(j = 1, \dots, N)$. In higher dimensions, this discretization will require a sophisticated mesh dealing with complicated geometries, which is the topic of Chapter 2.

High order polynomials are chosen for the approximation of the solution. The DGM uses the conventional Galerkin approach to solve the problem with compact support. In Chapter 3, the procedure will be applied for three dimensional Maxwell's equations.

By solving the DGM matrix system, the numerical results are obtained. Refinement strategies can be chosen according to the problem considered. Here, we compare h -refinement and p -refinement, which are performed by reducing the mesh size and increasing the polynomial order, respectively.

In Figure 1, the first results show a one dimensional cavity problem solved using a 1st order polynomial basis. The electric field values on the initial boundary are fixed as equation (1-3) shows. The field distribution of a sine function is represented with 4, 6 and 12 elements respectively. With time evolution, the one wavelength of electric field forms a standing wave with fixed boundary values. The elements are represented with 13 interpolation points for each of them. Unlike in conventional finite element methods, there is no continuity enforced at the interface of two neighboring elements. They are only connected through a numerical flux in a weak sense. The waves are shown at the end of one period of oscillation. The exact solution of the sine wave is shown in grey color for comparison with the series of solutions on h -refined meshes.

In Figure 2, the results are obtained by solving the problem with two elements using 1st, 2nd and 4th order polynomials. The results are displayed at the end of one period of oscillation. The field distribution is improved dramatically by p -refinement.

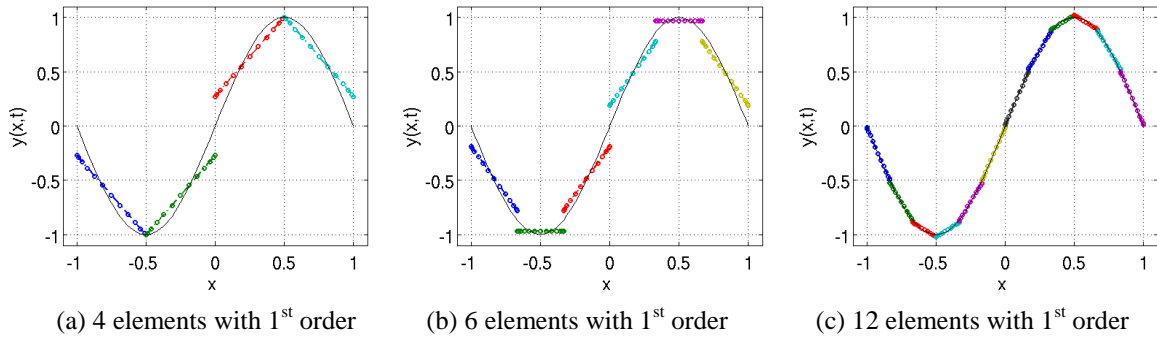


Figure 1: A one dimensional problem is solved by 4, 6 and 12 elements with 1st order polynomial basis.

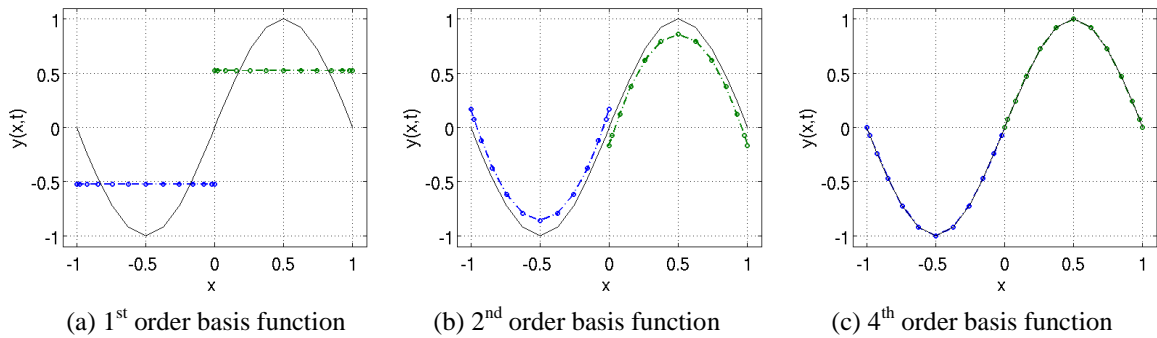


Figure 2: A one dimensional problem is solved by two elements with 1st, 2nd and 4th order polynomials.

The maximum errors of the above two approaches are given in Figure 3. On the left plot, initial interpolation errors using 1st order elements for a series of h -refined meshes are shown (in green color) and an exponential convergence with p -refinement (in blue color). On the right plot, the errors are at the end of one period of oscillation are shown. The zigzag curve shows that odd orders do not improve the accuracy. This is due to the symmetric shape of the sine wave when represented using two elements only as seen in Figure 2.

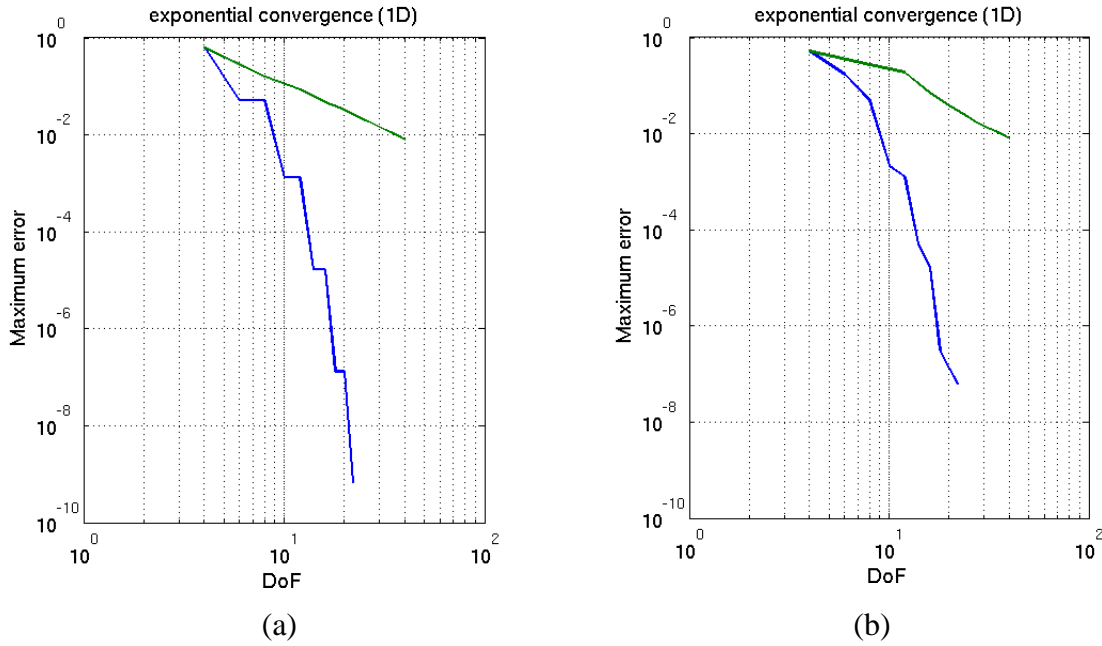


Figure 3: Exponential convergence in one dimension is depicted with the blue line for increasing the order, the green one for increasing the number of 1st order elements. The left side (a) plot shows the initial interpolation errors, while the right side (b) plot shows the errors measured after one period of oscillation.

1.1.2 High Order Geometric Description

The DGM can achieve exponential convergence as demonstrated in a one dimensional cavity problem above. However, the question remains how this convergence behavior can be obtained in higher dimensions, and what are the major restrictions. We motivate the topic by an observation of mesh refinements in a curved domain, and leave numerical results to Chapter 4.

Except for h - and p -refinement mentioned in the last section, a third option is the so-called r -refinement. The concept of r -refinement is invoked when mesh points are relocated such that they match the boundary of under resolved parts of the geometry. In this situation, the mesh resolution is raised with the same number of mesh points. It is recommended in the work [84] that p -refinement should be chosen when the solutions are smooth, and h -refinement is more efficient in regions with non-smooth solutions.

In Figure 4, we perform h -refinement on an initial mesh (a). The resulting mesh (b) is further improved by moving element vertices to the boundary. This leads to a global h -fitting mesh (c) with r -refinement.

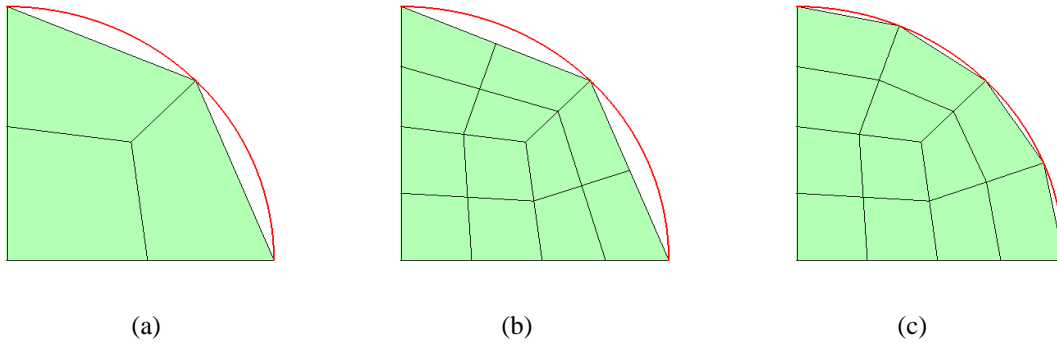


Figure 4: One quarter of a circular domain Ω (closed by a curved and two straight boundaries). Mesh (b) is obtained by h -refinement of mesh (a). Further r -refinement based on mesh (b) leads to a fitting mesh (c), where vertices of elements were relocated to the curved boundary.

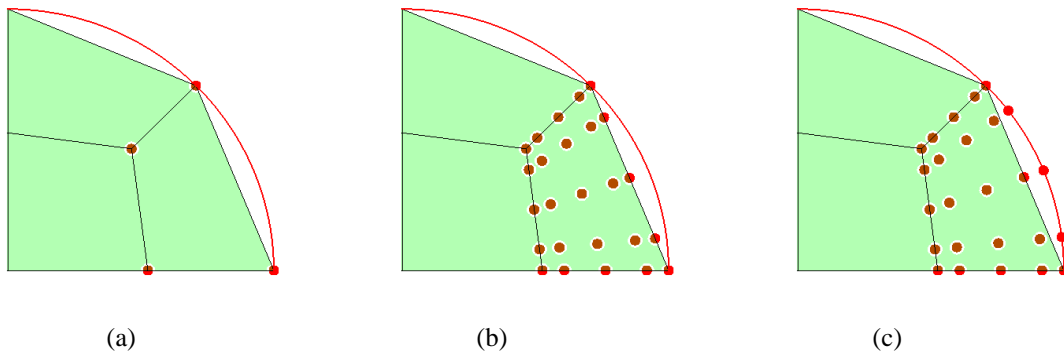


Figure 5: One quarter of a circular domain Ω (closed by a curved and two straight boundaries). Mesh (b) is obtained by p -refinement of mesh (a). Further r -refinement based on mesh (b) leads to a fitting mesh (c), where the interpolation points were rearranged on the curved boundary.

In Figure 5, we perform p -refinement on the mesh (a). The resulting mesh (b) is further improved by shifting the nodes onto the boundary. A p -fitting mesh (c) is, therefore, performed element-wise with local r -refinement.

It can be concluded that r -refinement is a suitable choice for objects with curved features, since the number of mesh points remains the same. The interpolation nodes are used for both, approximating the solutions and interpolating the geometry. The technique for h -fitting will be described in details in Section 2.1, and p -fitting in Section 2.3.

1.2. Overview

The motivation part has introduced previous related work, exponential convergence, as well as the three classes of mesh refinement strategies. The rest of the thesis is organized as the following.

In Chapter 2 a brief introduction of numerical methods and their mesh types is first given. Then a variety of mesh schemes are compared regarding to the mesh resolution on curved geometry features. After reviewing the quadrilateral mesh generation and refinement strategies, a body-conforming mesh scheme is presented. Based on the curved elements, the scheme is applied in numerical integrations to examine the accuracy in terms of geometric errors.

Chapter 3 describes the ingredients for high order finite element methods and introduces the hierarchical Legendre basis for approximating the solution. Then, a discretization of Maxwell's equations with the DG method is derived.

Chapter 4 provides details of the numerical results obtained from electromagnetic simulations. Convergence rates for the example of a cylindrical cavity problem are established. They show that with the application of the new mesh scheme high order accurate results are obtained also in the presence of curved boundaries. Additionally, an eigenvalue problem is solved via a time-domain simulation, and it is shown that highly accurate spectra free of spurious modes are obtained.

The thesis is concluded with a summary and future work in Chapter 5.

2. Body-fitting Mesh Scheme

Due to the lack of analytical solutions for real world problems, researchers usually resort to numerical computations. The starting point for most numerical techniques is the generation of a computational mesh, which divides the continuous domain of interest into a set of non-overlapping cells. Subsequently, the computation is performed in an element-wise fashion, where for example Maxwell's equations can be approximated by some technique within each cell. This process is called discretization. The most common numerical methods, based on volume discretization are the Finite Difference Method (FDM) [2, 12], the Finite Element Method (FEM) [13, 14], the Finite Volume Method (FVM) [15, 16] and, more recently, the Discontinuous Galerkin Method (DGM) [1, 17, 18, 19], which can be considered as a hybridization of the FEM and the FVM.

While the Finite Difference Method is typically applied on fully structured Cartesian meshes, the Finite Element Method is most often employed with tetrahedral or hexahedral meshes. The FVM can in principle be applied on arbitrary polyhedral meshes [20]; however, it is mostly applied on tetrahedral and hexahedral meshes due to the restrictions given by available mesh generators. The Discontinuous Galerkin Method inherits the flexibility regarding the choice of the mesh type from the FVM. However, unlike the FVM, the DGM is a high order accurate method. Ideal meshes for the DGM should, therefore, take the curvature of objects in the computation into account. To this end, the mesh elements themselves should be of high order, meaning that their boundaries described by explicit splines or implicit curves are known, which allows for fully exploiting the high order accuracy of the numerical method.

Despite the possibility of using various element types and ignoring the lack of mesh generators able of fulfilling the task of generating arbitrarily shaped elements, various properties of numerical methods are influenced by the actual quality of the mesh. More specifically the properties of the elements, for example their shapes, orientations and size densities [21, 22, 23, 24] have an impact on the numerical solution. The shape and orientation of the elements influence the stability and accuracy of the numerical computations, while the element size density is related with the efficiency of the simulations. Therefore, high quality mesh schemes should be designed to meet the following criteria:

- (1) Well-shaped elements. A mapping function transforms a reference element to the physical coordinates, where a well-shaped element can be defined by some indicators.

One of them is the so-called equiangle skewness, which measures the skewness of an element [25]. It is defined by

$$S = \max \left\{ \frac{\theta_{max} - \theta_e}{180^\circ - \theta_e}, \frac{\theta_e - \theta_{min}}{\theta_e} \right\}, \quad (2-1)$$

where θ_{max} and θ_{min} are maximum and minimum angles between the edges of elements, and θ_e is the characteristic angle (60° for triangle and tetrahedral, 90° for quadrilateral and hexahedral). An element is less skewed when S is closer to 0. Another indicator is aspect ratio, which is the length ratio between the longest and shortest edges. An aspect ratio of one is the best case.

- (2) Geometry-oriented elements. This criterion presents how flexible a mesh scheme deals with controlling orientations of the elements. It is desirable to have edges (faces) of the elements aligned with the local feature of material interfaces as well as domain boundaries [21].
- (3) Element size density. The element size density over the domain is defined by the mesh size control, which depends on the global mesh, local features, size gradation and some predefined functions [26]. The element size is initialized by the background mesh, and should later be adapted to the local size and curvature of the geometry. The size gradation means gradual variation of element size, which can be guided by a refinement strategy based on quadtrees (octrees) with hierarchical refinement levels. This strategy is detailed in Section 2.1.2. The refinement process can also be guided by a scalar goal function as proposed in [22, 27]. This function can drive the generation of body-fitting meshes taking into account properties such as curvature, local feature size or user-specified functions.

The aim of this chapter is to develop a high order mesh scheme. We first introduce basic terminology and notation such as mesh topologies and element types. In the following, a mesh design algorithm is derived, which takes the above described criteria into account. In Section 2.1, several schemes are demonstrated and compared with regard to their abilities of generating meshes for geometries that include curved features.

We focus on a quasi-structured mesh scheme [28] in Section 2.1.3 with quadrilateral (hexahedral) elements. Section 2.2 covers different options of boundary descriptions, such as implicit form or explicit spline form. In this section, a technique is introduced to simplify the generation of meshes, which have the form of a structured Cartesian mesh in large regions and apply a so-called buffer layer for connecting to the boundary. Validation of the mesh scheme is performed with curved boundaries in both implicit and explicit forms.

Section 2.3 further presents a procedure to fit the geometry element-wise with high order accuracy. It is demonstrated that the generated meshes, indeed, fit curved boundaries with high order accuracy. To this end, it is shown that the error in the representation of the domain area converges at a high rate if high order curved elements are employed. In

Section 2.4, the mesh scheme is extended into three dimensions. A variety of mesh examples will be presented throughout this chapter.

2.1. Mesh Generation for Curved Geometries

For the generation of meshes suitable for the computation with high order Finite Element Methods, various degrees of freedom exist. These include element specific properties such as the different element types, which can be triangular (tetrahedral), quadrilateral (hexahedral) or other polygonal (polyhedral) elements, as well as global mesh properties such as topology. Meshes can be fully structured (as purely Cartesian meshes), fully unstructured and quasi-structured, which is the approach followed in the thesis. These terms will be clarified later in this section.

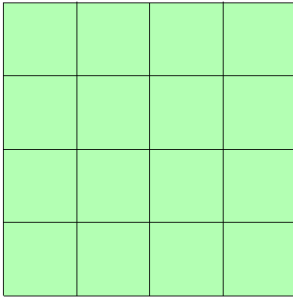
2.1.1 Background Mesh Generation

The process of generating body-fitting meshes using high order curved elements follows several steps. The first step is the creation of a so-called background mesh, which describes the basic underlying mesh structure. A Cartesian mesh, as the simplest background meshes, can be efficiently generated over the whole domain. It is composed of quadrilateral or hexahedral elements. For demonstration purpose, until Section 2.3 two-dimensional meshes are explained in details. Then in Section 2.4 the extension of the mesh algorithms to three-dimensional space is described.

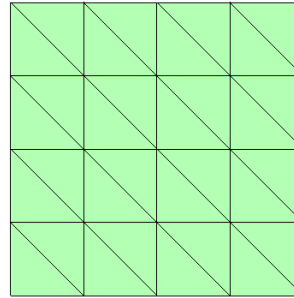
As a starting point, a Cartesian mesh is defined with uniformly distributed elements. A widely used alternative is a fully unstructured mesh composed from triangular elements of varying size. The other possible choices are structured mesh with triangles and unstructured mesh with quadrilaterals respectively. However, only the former alternative is considered in the following and the latter ones are mentioned for completeness only. A comparison of the four background meshes mentioned is illustrated in Figure 6.

In the structured meshes depicted in the Figure 6(a) and (b), every interior mesh point always has a constant number of neighboring elements. Once the nodal points of a structured mesh are generated, the connectivity of the elements is established as well as a direct consequence. With the index of a point (i, j) in a two dimensional Cartesian mesh, one can easily find a triangle or a quadrilateral element by the indices set $\{(i, j), (i + 1, j), (i, j + 1)\}$ or $\{(i, j), (i + 1, j), (i + 1, j + 1), (i, j + 1)\}$, respectively. Together with the index of the boundary points, element connectivity is uniquely defined by a certain periodic pattern. For instance, structured meshes divide a square domain into quadrilateral elements in (a) and triangular elements in (b). If we connect an interior element and its

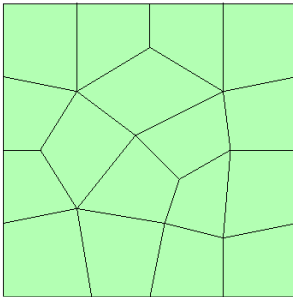
neighbors by their centers, one will find a “cross-shaped” and a “Y-shaped” pattern in (a) and (b), respectively. The periodic distribution of such a local pattern determines the element connectivity globally.



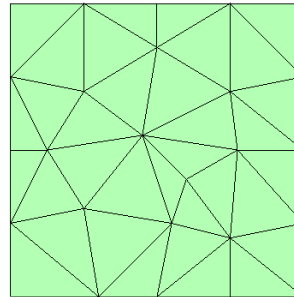
(a) Structured mesh with quadrilaterals



(b) Structured mesh with triangles



(c) Unstructured mesh with quadrilaterals



(d) Unstructured mesh with triangles

Figure 6 : A square domain Ω (closed by straight boundaries) is a special case where the approximated domain Ω_h introduces no geometric error. The different meshes can be described in two ways: structured (a, b) and unstructured meshes (c, d), or quadrilateral (a, c) and triangular element meshes (b, d).

Unstructured meshes can be generated with iterations of adding, deleting or adjusting the position of mesh points. In an unstructured mesh, the number of neighboring elements can vary from node to node. As examples, the square domain of Figure 6 is divided into (c) quadrilateral and (d) triangular elements in way that clearly leads to unstructured meshes. Due to the complex topology of the final mesh, a periodic pattern of element connectivity is no longer available. Consequently, the elements vertices and their connectivity have to be stored explicitly. Given a specific element, its neighboring elements are found with an element connectivity table. For complicated real-world problems the connectivity table can contribute significantly to the required memory but also to the computing time if it has to be parsed repeatedly. The latter case would occur if freely moving particles were to be tracked or if adaptive mesh refinement is applied.

Note that the square domain of Figure 6 is an example, where the discretization of the domain of interest using simple straight-sided elements does not lead to any geometric error.

2.1.2 Mesh Schemes for Curved Boundaries

As stated in the beginning of this chapter, the quality of the mesh is a major concern. Different schemes for the accurate treatment of curved boundaries will be introduced and compared. We concentrate on the properties of elements in terms of shape, geometric-orientation, and size density.

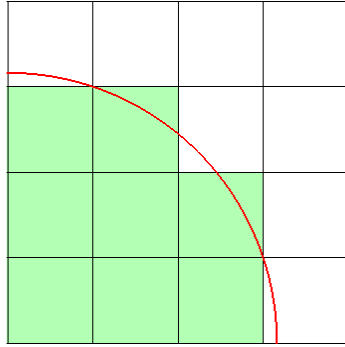
Four mesh strategies are described in Figure 7 for one quarter of a circular domain Ω . The boundary $\partial\Omega$ consists of one curved and two straight segments. Since the meshes coincide exactly with the straight part, the geometric errors are only introduced by meshing the curved segment.

Triangular meshes are the most popular choice for complicated geometry. Strategies frequently used are the Advancing Front Method [29, 30, 31, 32, 33] and the Delaunay Method [34, 35, 36, 37, 38]. Some hybrid methods are also available: a study in [39] combines the Delaunay and the octree ideas; a method in [40] takes advantage of Delaunay and Advancing Front Method. In order to generate meshes these methods insert, eliminate and modify a large number of points. Attention has to be paid to make sure that the mesh is still valid after numerous operations. We will concentrate on a mesh scheme with quadrilateral elements.

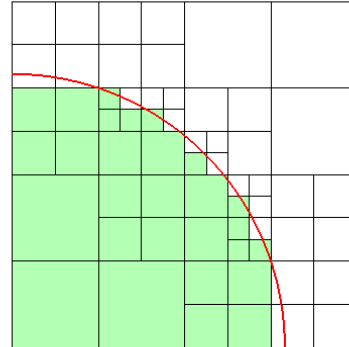
Four possible strategies are outlined, which are visualized in Figure 7. The first strategy is the simplest one. Here, a fully structured Cartesian mesh is employed, which corresponds to the above introduced background mesh. This strategy clearly leads to a poor staircase-like representation of curved objects and boundaries. In Figure 7 (b) the background mesh is refined employing a so-called quadtree refinement [41, 42, 43, 44]. Elements cut by a boundary or interface are iteratively split into smaller elements. As the splitting occurs, four equally sized elements emerge from one parent element, which is referred to as quadtree refinement. In three-dimensional space the respective term is octree refinement accounting for the eight children of one parent element. Although the boundary representation maintains its staircase character, the representation error is eventually drastically reduced. Quadtree refined meshes have the advantage of improving the boundary representation while keeping element edges or faces strictly aligned with the Cartesian coordinates, which is beneficial in conjunction with the DGM to be applied on these meshes. Quadtree refined meshes can be seen as a background mesh, which is locally improved in a way that it maintains Cartesian without a global refined mesh throughout the domain. Despite its advantages over the fully structured Cartesian mesh, the staircase boundary remains a significant drawback, especially in the context of high order numerical methods.

The strategies three and four, depicted in Figure 7 (c) and (d) give up the alignment of elements with Cartesian coordinates. While the mesh shown in (d) can be considered a standard mesh for FEM simulations, the quadrilateral mesh of (c) achieves the same error

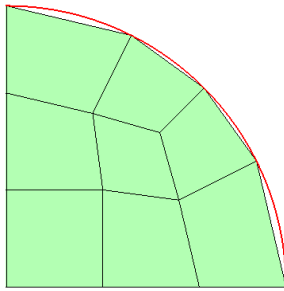
in representing the curved segment while employing half the number of elements. Quadrilateral and hexahedral elements, i.e. tensor product type elements in the general case, have a number of other advantages over triangular/tetrahedral elements when applied with the DGM. These will be detailed in Section 3.4.2.



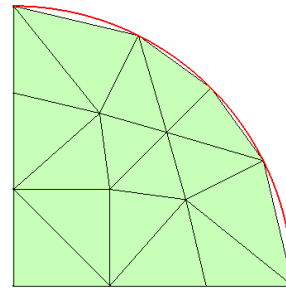
(a) Cartesian mesh



(b) Quadtree mesh with hanging nodes



(c) Unstructured mesh with quadrilaterals



(d) Unstructured mesh with triangles

Figure 7: One quarter of a circular domain Ω (closed by a curved and two straight boundaries) is approximated using four mesh schemes: (a) Cartesian, (b) Cartesian with quadtree refinement and (c, d) unstructured meshes. The approximation to the curved boundary is composed by the union of boundary edges of the elements, which are either (a, b) piece-wise constant, or (c, d) piece-wise linear.

In the following the meshes depicted in Figure 7 are evaluated with regard to the criteria given above. According to equation (2.1), the values of the equiangle skewness S are all zero for elements in the Cartesian and quadtree mesh. These square elements are always well-shaped. The resulting discretized geometry has a staircase boundary. Therefore, edges of elements are in general not well aligned with interfaces and boundaries.

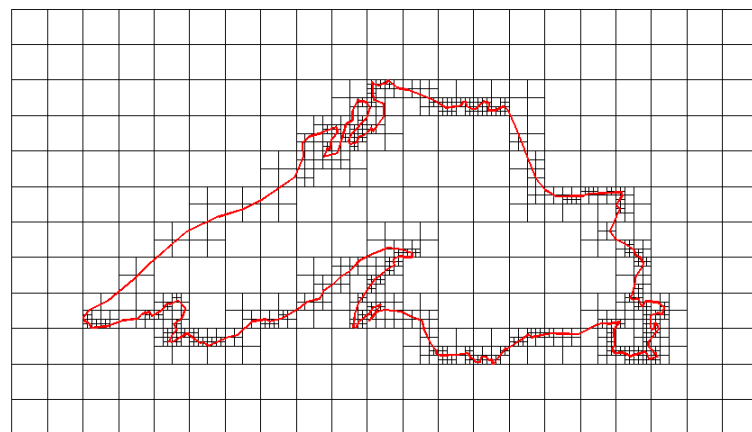
The element skewness for the unstructured meshes depends on a suitable placement of mesh points. It is, however, well-known that this is a serious issue if more complicated structures are to be meshed [24].

Regarding element size density, all strategies but the quadtree one perform well, yielding elements of similar sizes. However, this is also known to be an issue for more complicated structures, with the exception of the structured Cartesian mesh. In the upcoming sections, a combination of quadtree refined meshes, which are locally unstructured will be developed. This allows for a compromise in the sense that many problems that come along with the generation of globally unstructured meshes with possibly curved elements are avoided by using quadtree refinement to make the meshing process more localized. This localization effect is visualized in Figure 8, where a mesh for a complicated structure is generated. Figure 8 shows the curved boundary of a lake domain is discretized by a quadtree mesh.

The central question is at which location the mesh should be refined. Since we consider geometric errors in this work, the important criterion are the small local features on the boundary, i.e. the coordinates of the critical points on the curved parts of the geometry. Guided by these critical points, the algorithm performs the quadtree refinement. The density of the critical points determines the number of refinement iterations.



(a) Elements inside and outside the boundary are selected from a Cartesian mesh.



(b) Elements on the boundary are refined by a quadtree mesh.

Figure 8 : A lake domain with curved boundary is discretized by a quadtree mesh. The small details of the boundary are captured by the refined mesh.

In Figure 8(a), the elements that do not cover the border are selected from the background mesh. We have chosen some critical points on the boundary. According to their position in the background mesh, a flexible number of iteration times is set, meaning the closer a critical point is to the border of an element, the more quadtree refinement iterations are likely to be required in order to achieve a good approximation.

Figure 8(b) depicts that the small details of the border are captured by the refined mesh. It should also be noted that the quadtree strategy is the only one out of the four that have been introduced, which creates so-called non-conforming meshes. In other words, quadtree refinement leads to non-unique neighborhood relations, where one element does not necessarily have one neighbor but might have several of them. The mesh nodes on such non-conforming edges are called hanging nodes as there is one direction along which no edge is connected to the node. As the DGM can cope with non-conforming meshes this is no essential concern.

In order to overcome staircasing for Cartesian and quadtree meshes, a template based refinement will be presented in the next section. Other efficient strategies for quadrilateral mesh generation will also be considered.

2.1.3 All Quadrilateral (Hexahedral) Meshes

As mentioned before, quadrilateral or more general tensor product type meshes have a number of advantages over fully unstructured triangular or tetrahedral meshes. The most obvious one being that significantly fewer elements are required for covering a given domain of interest. More specific advantages in the context of the Discontinuous Galerkin Method will be addressed in Chapter 3. However, the former advantage alone already makes it worth investigating tensor product meshes.

For the generation of meshes, tensor product-type or tetrahedral meshes as well, two most commonly used approaches exist. They differ in the starting point for the construction process, where one option is to grow a mesh starting from the geometry boundary and the second is to begin with a regular background mesh. According to this first step, it is convenient to classify quadrilateral (hexahedral) mesh generation into two categories: geometry-first and mesh-first [21, 23].

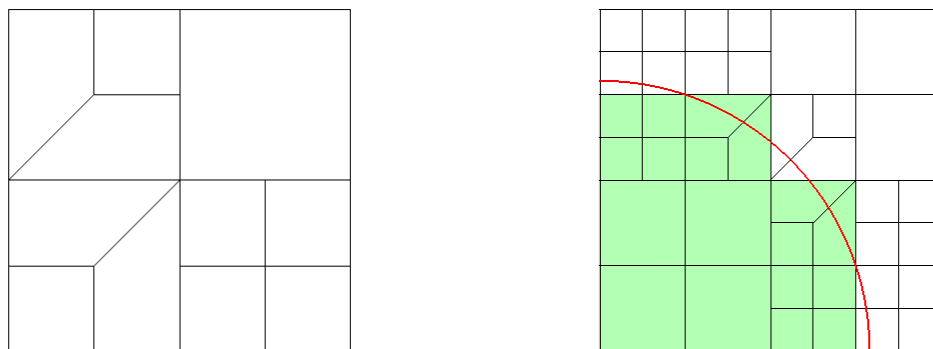
(1) Geometry-first. In this approach the starting point is a parameterized boundary of the geometry. Elements are created along the boundary first. Afterwards, the mesh is grown progressively towards the inside of the domain until it is fully covered. A typical approach is the advancing front method [21, 24, 45, 47, 48, 49]. Concerning the list of quality measures given at the beginning of Chapter 2, one can say that with such a method, the orientation of the element is easy to control. Local refinement is supported to reduce mesh size. As the mesh is grown from the boundary, usually

highly distorted elements are not a concern but they tend to be well defined. However, one of the drawbacks is that problems can occur towards the center of the mesh, where different advancing fronts have to be matched. This sometimes results in a lower quality of the interior elements.

- (2) Mesh-first. For this approach the beginning is the kind of fully structured mesh that has been introduced above as the background mesh. Though a background mesh is usually a fully Cartesian mesh, a refined quadtree (octree) mesh can serve as an intermediate step towards a real body-fitting mesh as well. Refining the structured mesh using quadtrees simplifies the capturing of geometry features as they are embedded to some extent into the background mesh already. Representative strategies for mesh-first methods are found in [23, 45, 46, 48, 49]. If a mesh-first strategy is followed the interior elements are usually well-shaped and the challenge is to generate geometry-oriented elements with high quality on the boundary.

Here, we follow the second approach and construct a mesh scheme based on the mesh-first idea. Four approaches to obtain a mesh with a good geometry representation are illustrated in Figure 10, where again a quarter of a circular domain Ω is used for illustration. All approaches refer to Figure 7, where in (a) the background mesh (with no quadtree refinement) of is the starting point. In Figure 10 (a) the mesh is simply refined globally by a factor of two. While improving the geometry representation, the problem of staircasing remains. Also this strategy quickly leads to a very large number of elements and is therefore dismissed. The other three strategies for turning the background mesh into a body-fitting mesh are shown from Figure 10 (b) to (d). Each of them is described in the following.

Specifically, two types of refinement templates will be uses, the quadtree and “Y-shaped” template, which are shown in Figure 9(a).

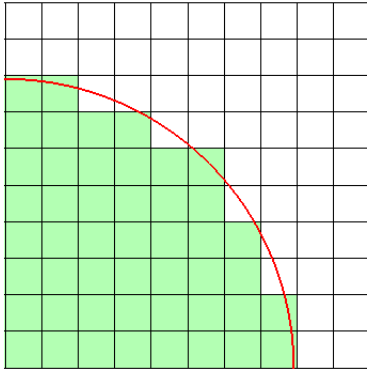


(a) The left two elements are refined with “Y-shaped” template. The upper right element is unrefined, and the lower right element is refined with quadtree template.

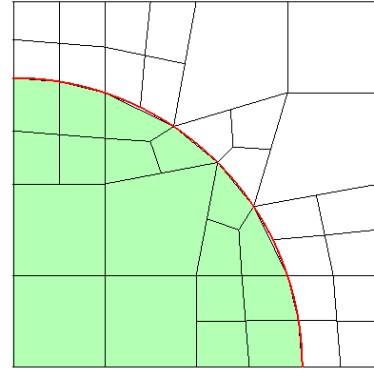
(b) The elements are refined with the “Y-shaped” template when they have three vertices on the same boundary.

Figure 9 : Refinement templates and their refined mesh for one quarter of a circular domain before the selected elements (green color) are deformed.

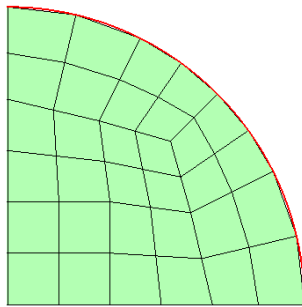
The “Y-shaped” refinement divides an element into three sub-elements, which can easily be fitted to a body by moving the interconnecting node without causing degenerated quadrilaterals. Therefore, it is suitable to control orientations of the refined elements along the boundary. Figure 9(b) depicts the refined mesh, which is one step before the boundary fitting. A further deformed mesh is shown in Figure 10 with other refine strategies.



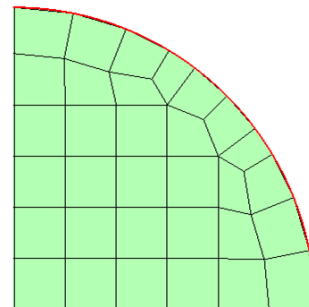
(a) The Cartesian mesh is refined by reducing the mesh size globally.



(b) The mesh is refined along the curved boundary with templates.



(c) The unstructured mesh in Figure 2(c) is refined by dividing each element into four.



(d) The quasi-structured mesh is based on a refined Cartesian mesh.

Figure 10 : One quarter of a circular domain Ω (closed by a curved and two straight boundaries) is covered by quadrilateral meshes: (a) Cartesian mesh (b) Template refined mesh (c) Unstructured refined mesh (d) Quasi-structured mesh.

The mesh of Figure 10(b) is obtained by a strategy, which is somewhat similar to quadtree refining mesh shown in Figure 7(b). In the latter case, the refinement is carried out twice with a quadtree template. Here, refinement is carried out once only making use of both, the quadtree and the “Y-shaped” options. The latter one is applied to elements, which have one node that is reasonably close to a boundary. In this case, the respective node is shifted such that afterwards it is located exactly on the boundary. In a second step, quadtree refinement is applied to elements, which have two nodes located on the

boundary in order to make the connecting edge shorter. The extra node is positioned on the boundary as well, yielding a significantly improved representation of the curved segment. Using this additional flexibility, the refined mesh in Figure 10(b) achieves a reasonable representation of the curved arc using a rather small number of elements. On the downside, the mesh size density is the poorest of the four options. Moving element nodes also immediately affects the equiangle property of an element. In general situations it can lead to a number of rather heavily distorted elements.

The mesh depicted in Figure 10(c) is obtained by refining once each element of the unstructured mesh of Figure 7(c). In the context of mesh adaptation this corresponds to global h -refinement. For this mesh the elements are well aligned with the boundary, the equiangle skewness is well-behaved and the elements all have very similar sizes. While all this speaks in favor of using fully unstructured meshes, it is at the same time the weak point. Though unstructured meshes can be handled with the DGM, it strongly impacts the computational costs of the method, as will be detailed in the following chapter.

Finally, a fourth strategy is illustrated in (d). The mesh is based on a mesh-first approach [45]. As a result, it succeeds at representing boundaries using well-shaped, boundary-aligned elements of similar size in a so-called buffer layer and maintaining a structured Cartesian mesh throughout large parts of the mesh. Following the definition of Section 2.1.1 the mesh is structured almost everywhere, with the exception of a number of nodes at the interface of the buffer layer with the interior Cartesian mesh as can be seen in Figure 10(d). The resulting mesh is called a quasi-structured mesh, which in this case has the same number of element as the one of Figure 10(c). If curved domains or objects are to be considered, only the elements near the boundary have to be curved.

Figure 11 illustrates how the strategy based on these templates can cope with the problem of generating body-fitting meshes for a circular disc being embedded into a square domain, where the size of disc is varied in six steps. Resizing the disc changes the problem fundamentally as with increasing size an interface has to be represented by the mesh, whereas in the first step shown in Figure 11(a) a pure embedding problem has to be considered. There are four levels of hierarchically refined elements with three refinements in total. The quadtree template is applied exclusively for all levels of refinement, except the last one. Special attention has to be paid to keep all the elements well-shaped when associating the vertices of elements on the boundary.

The goal is to examine, whether the algorithm produces valid meshes of good quality. An inspection of Figure 11 shows that this goal is met with the usual limitations of the template based refinement, i.e. varying element sizes.

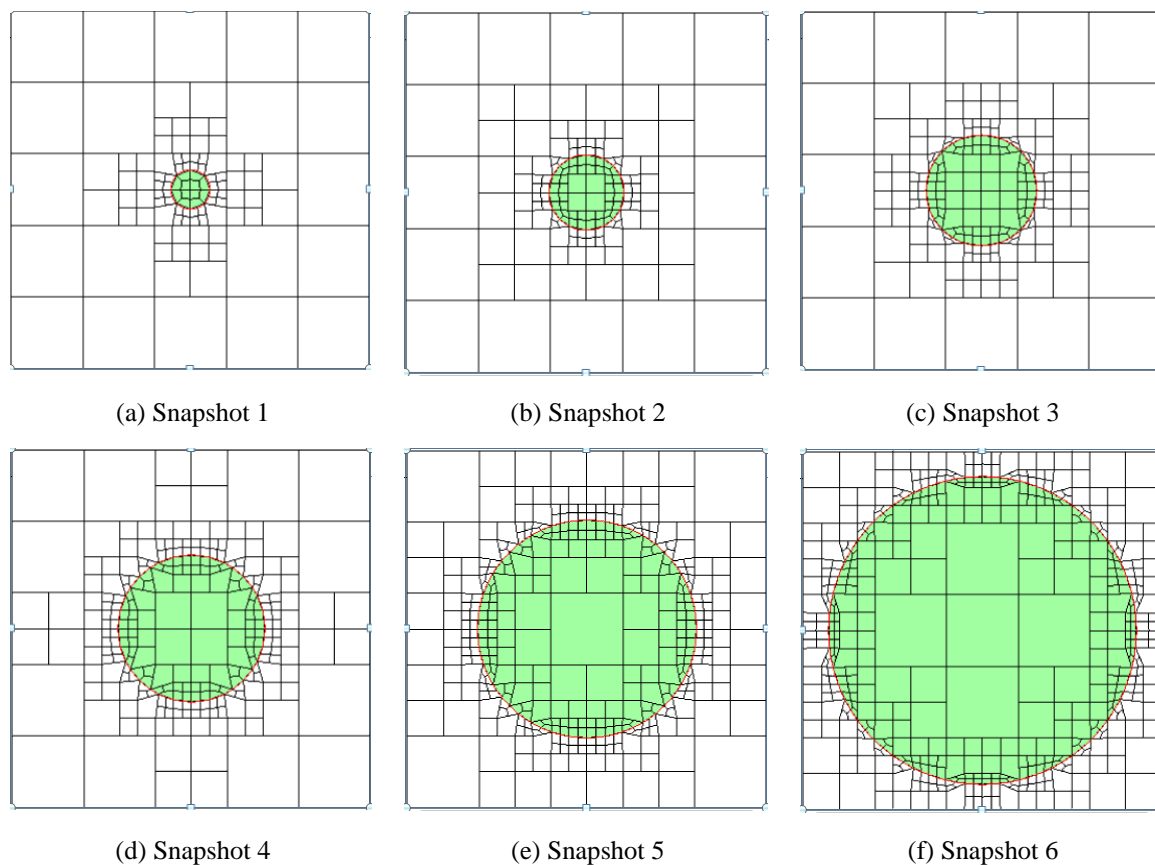


Figure 11 : Mesh generation of a moving curved boundary in a square domain. The quadtree and “Y-shaped” template refinements are carried out at the interface around the curved boundary.

2.2. Implicit and Explicit Form of Boundaries

The mesh generation for complicated geometry needs information about the quantities of interest on the boundaries. These quantities are commonly represented with boundaries that are defined in implicit and/or explicit form. Due to the arguments laid out in the previous section the quasi-structured mesh scheme generates the best meshes for our purpose. In the following its details are presented for boundaries given in both forms.

The input data for an implicit geometry can be a continuous functions or points cloud, on which the boundaries are described. In two dimensions, the zero level set describes the boundary with a contour line $f(x, y) = 0$ or contour points $\{(x, y)\}$. For instance a circle $f(x, y) = x^2 + y^2 - 1 = 0$ in Figure 12(a) can provide a signed function, which tells for any point in the domain if it is on the boundary ($f = 0$), inside ($f < 0$) or outside ($f > 0$) the boundary. A mesh generator for implicit geometries can be found in [27], where it is applied to problems in fluid dynamics and structure deformation.

A common explicit form for describing geometries are Non-Uniform Rational B-Splines (NURBS) [50, 53], which have become the standard in most popular Computer Aided Design (CAD) and Computer Aided Engineering (CAE) systems.

NURBS are constructed from B-splines, which are a generalization of Bezier curves [54] to rational functions. In particular, a circle can be represented exactly using NURBS as shown in Figure 12(b). Another example of NURBS will be given after the introduction of B-splines.

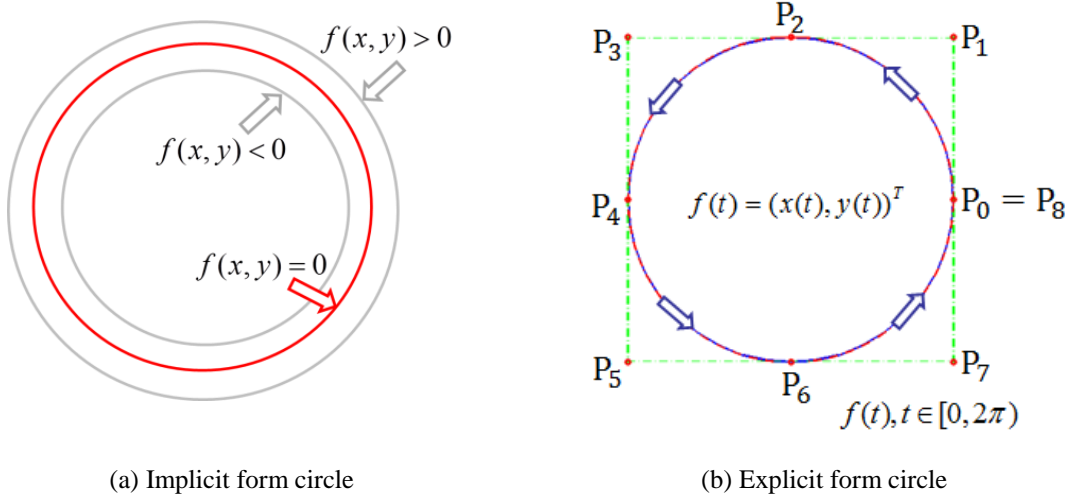


Figure 12 : A circular boundary $\partial\Omega$ is represented by implicit and explicit forms. (a) The implicit form uses a signed function $f(x, y)$ to tell whether a point is on the boundary, while (b) the explicit one relies on NURBS to parameterize the boundary where red nodes are the control points for the basis functions.

The construction of B-splines requires a given knot vector as a set of non-decreasing coordinates in the parametric space. It is written as $\Xi = \{\xi_0, \xi_2, \dots, \xi_m\}$, where ξ_i is the i -th knot, and $(m + 1)$ is the number of knots. Given a set of control points $\{P_0, P_2, \dots, P_n\}$ then, $k = m - n - 1$ is called the polynomial order (or curve degree). The knots $\{\xi_{k+1}, \xi_2, \dots, \xi_{m-k-1}\}$ are called internal nodes. They are the main difference between B-spline and Bezier curves, as we will see for the example of a circle later. A B-spline curve of degree k is defined by:

$$C(\xi) = \sum_{i=0}^n N_{i,k}(\xi) P_i, \quad (2-2)$$

where the geometric coefficient P_i is the i -th control point. The basis function $N_{i,k}$ is defined recursively as:

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise,} \end{cases} \quad (2-3)$$

and for $k \geq 0$ as:

$$N_{i,k}(\xi) = \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} N_{i,k-1}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} N_{i+1,k-1}(\xi). \quad (2-4)$$

A B-spline curve can be modified and designed by specifying a number of control points. It is well known that only rational functions can represent conic curves. As a generalized form of B-splines, NURBS are defined as a weighted rational function. A knot vector of NURBS is not required to be equally spaced, in other words, it is in general non-uniform. Using the same Bernstein functions, a NURBS curve takes the form:

$$C(\xi) = \frac{\sum_{i=0}^n w_i N_{i,k}(\xi) P_i}{\sum_{i=0}^n w_i N_{i,k}(\xi)}, \quad (2-5)$$

where control point P_i and basis function $N_{i,k}$ are defined before. A new scalar w_i is introduced, which is called the i -th weight. When the weight w_i is varied in the process of a curve design, its corresponding control point P_i will “attract” or “repulse” the curve [53].

A very convenient application of NURBS is the representation of conic curves, which can be represented exactly. To this end, several circular arcs are jointed together by taking the last point of the current arc as the first point of the next one. In the case of a quadratic order, one can choose $(2x + 1)$ control points, where x is the number of arcs. A possible choice is shown in Figure 12(b), where a circle is defined by four arcs with nine control points $\{P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$ (red nodes). The sequential connection of the control points is called the control polygon, which in this case is a square depicted by a green dashed line. The resulting circle can be considered either as a piecewise rational Bezier curve or as a NURBS curve [53]. When no internal knots exist, a B-spline degenerates into a Bezier curve [51]. In such a case, B-spline basis functions coincide with n -th degree Bernstein polynomials. To be specific, no internal knots means that a knot vector has only values of zero in the first $(k + 1)$ knots and one in the last $(k + 1)$. More details about NURBS can be found in [52]. As a simple approach of quadratic NURBS, the knots for a circle can be chosen as $\Xi = \{0,0,0,1,1,1\}$, which has the same form as the Bezier curve. The coordinates of the control points and their weights are given in Table 1, where even indices of the control points have weights of $\frac{\sqrt{2}}{2}$, and the odd ones have weights of 1.

Table 1 Coordinates and weights of the nine control points for a NURBS approach to represent a circle

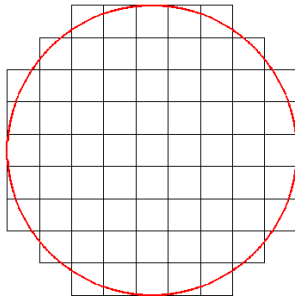
Control points	x -coordinate	y -coordinate	weight
Point P_0	1	0	1
Point P_1	1	1	$\sqrt{2}/2$
Point P_2	0	1	1
Point P_3	-1	1	$\sqrt{2}/2$
Point P_4	-1	0	1
Point P_5	-1	-1	$\sqrt{2}/2$
Point P_6	0	-1	1
Point P_7	1	-1	$\sqrt{2}/2$
Point P_8	1	0	1

2.2.1 Curved Boundaries with the Buffer Layer Scheme

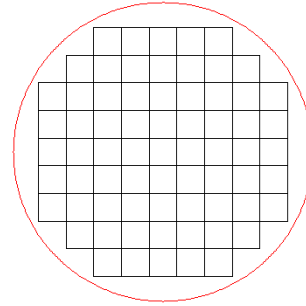
Generating a body-fitting mesh starting from a background mesh is a difficult task due to a number of issues. Elements have to be reoriented in order to control their alignment with the prescribed boundary (surface) of the geometry. This task includes the issue of shifting element nodes to boundaries and surfaces. Specifically the problem is found in avoiding element twists, where two nodes interchange their order during the shifting process creating an element twist. The solution is found by exploiting isomorphisms as described in [45]. The basic principle of isomorphic meshes reads: for each node $\{N_1\}$ on the surface of an internal polygon, there is a unique node $\{N'_1\}$ on the boundary; for each edge $\{N_1N_2\}$ on the surface of a polygon, there is a unique edge $\{N'_1N'_2\}$. Therefore, a unique quadrilateral element $\{N_1N_2N'_1N'_2\}$ is defined by an isomorphic mesh. This states that it is indeed possible to construct a valid mesh.

We start with the Cartesian background mesh. One possible and promising solution is to apply buffer layers [21, 28, 45] as shown in Figure 10(d). The buffer layer is an example of an isomorphic mesh region, which is inserted between curved boundaries and the interior structured mesh.

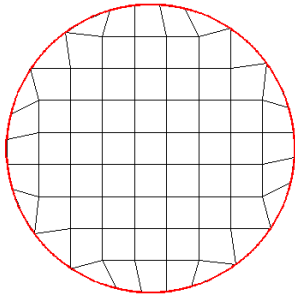
We will quickly describe what happens, if the isomorphism property is violated. In Figure 13(a) a part of the background mesh is selected and transformed to be body-fitting in Figure 13(b). During this process, the so-called “doublet” occurs, when the same element has two neighboring edges to be fitted on the boundary. The common vertices of the two edges are marked with black circles as “doublets” in Figure 13(c). Though the boundary representation is good, the problem with a doublet is that it is likely to produce a degenerate element, meaning that the angle of the two edges is approaching 180 degrees. In this case the distinction between the two coordinates is lost. As a consequence the determinant of the Jacobi matrix at the position of the node is zero or negative.



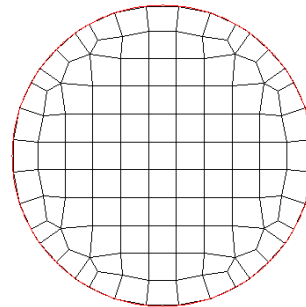
(a) Construct a “staircase” mesh by choosing elements whose center is inside the circle



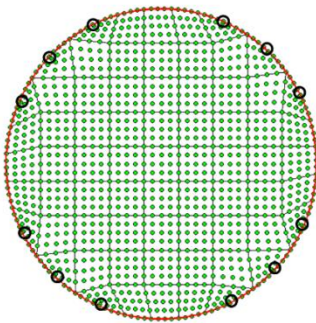
(d) The mesh size is smaller with the same number of elements as (a). All vertices are inside the circle.



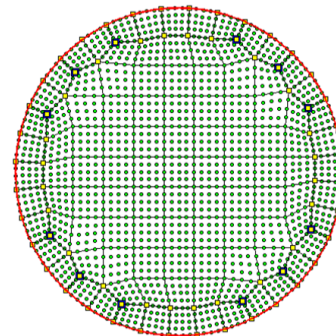
(b) Associate vertices of elements directly to the boundary



(e) Parameterize the boundary with points to insert buffer elements



(c) The doublets in the degenerated elements are marked with black circles.



(f) The degenerated elements in (c) are recovered to normal quadrilaterals with a set of buffer elements.

Figure 13 : A circular domain Ω is tessellated with elements according to two mesh schemes: (c) twelve degenerate elements are produced near the boundary, while in (f) geometry-oriented and well-shaped elements are obtained.

The issue of “doublet” generation is solved by the insertion of a buffer layer, which guarantees non-degenerate well-shaped elements [21]. The steps to generate buffer layer meshes are illustrated in Figure 13(d)- (f), where a set of elements is filled in the layer between the exact curved boundary and the interior structured mesh.

Figure 14 shows a few other examples of simple curved domains discretized by the buffer mesh scheme. The elliptical domain (a) and the cross section of a coaxial cable (b) are described by implicit boundaries. For an explicit case, we describe a deformed circle with parametric NURBS boundary in Figure 15. For all three examples the algorithm succeeds in generating a valid mesh consisting of well-shaped elements that are fitting the curved boundary. Especially, the NURBS form is able to represent and modify the boundary. Examples will be given in the following section, where NURBS are utilized to transform implicit contour points into parametric boundary, as well as design and modify high order curved boundaries.

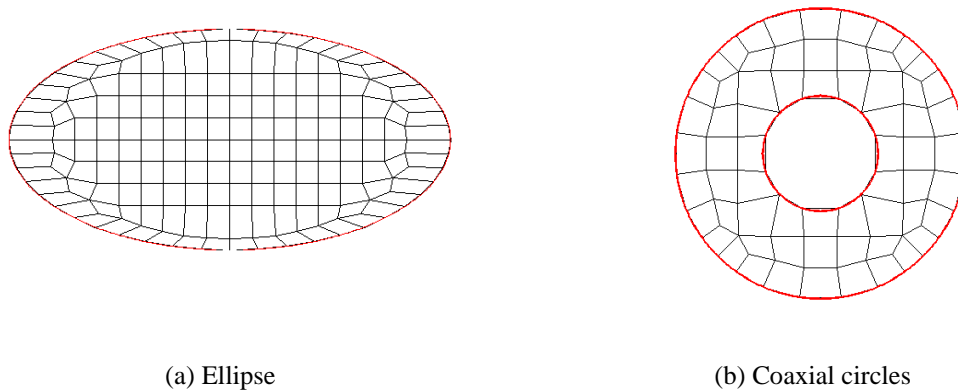


Figure 14 : Meshes are equipped with buffer layer elements. Two examples of implicit forms are considered: (a) an ellipse and (b) a coaxial cable.

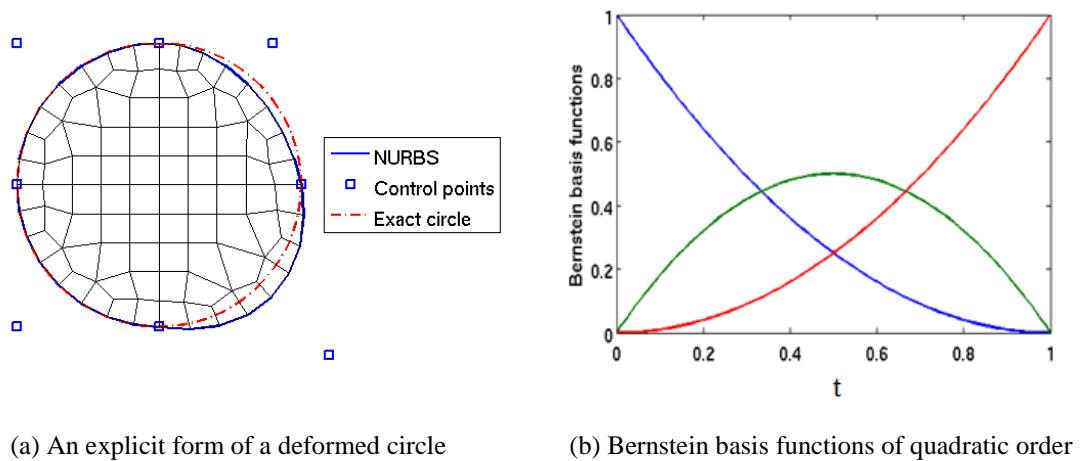


Figure 15 : Buffer layer elements and basis functions for an explicit form of a deformed circle.

One essential question for any mesh generator is how to deal with arbitrary and complicated geometries. We improve the buffer layer scheme in the next section by incorporating a distance function and show several applications, which show the applicability in more complicated situations.

2.2.2 Applications with Buffer Elements

The generation of buffer elements requires the determination of points at the inner and outer interface of the buffer layer. The inner interface shares the points with the interior structured mesh. The outer interface coincides with the boundary. The interior points are obtained easily by extracting the subset of elements of the background mesh, which do not touch any interface or boundary. This is depicted in Figure 16(b), Figure 17(b) and Figure 18(b). Obtaining the respective points on the outer interface requires a projection of the inner points onto the boundary or interface. For most geometries with moderately curved boundaries, the insertion of one layer of elements is enough for buffering the mesh deformation and a straightforward projection using the inward normal of the boundary can be done. For more complicated problems, however, it can be difficult to obtain the exact positions of the projected points on an arbitrary curved boundary, especially when the mesh is rather coarse due to, e.g., twisting effects. In this case, a useful tool is an approximate projection, as proposed by Persson in [61].

First the calculation of an exact projection is described. Given an interior interface point \mathbf{p} and an implicit function $\phi(\mathbf{p})$ and assuming an isomorphic situation, there is a boundary point $(\mathbf{p} + \Delta\mathbf{p})$ that satisfies the zero level set

$$\phi(\mathbf{p} + \Delta\mathbf{p}) = 0. \quad (2-6)$$

Here, $\Delta\mathbf{p}$ describes the vectorial distance from the interior interface point \mathbf{p} to the point on the boundary. It is found by projecting the interior point along the boundary normal. The problem with this approach is that this projection can fail to produce a valid mesh in situations where the boundary is corrugated and the background mesh is coarse. This is remedied by replacing the exact projection with a distance function. Following this approach, the interior points are not projected to the boundary orthogonally but placed at the boundary location with the smallest distance from the point under consideration.

To this end, a scalar factor t is introduced, which allows for expressing the distance vector $\Delta\mathbf{p}$ as a vector that is parallel to the gradient at the boundary point $(\mathbf{p} + \Delta\mathbf{p})$, rather than at the point \mathbf{p} :

$$\Delta\mathbf{p} = t\nabla\phi(\mathbf{p} + \Delta\mathbf{p}). \quad (2-7)$$

The first order approximation of the projection is a truncation of the Taylor expansion of the implicit function ϕ at point \mathbf{p} :

$$\phi(\mathbf{p} + \Delta\mathbf{p}) \approx \phi(\mathbf{p}) + \nabla\phi(\mathbf{p}) \cdot \Delta\mathbf{p} \quad (2-8)$$

Assuming that $\nabla\phi$ is evaluated at point \mathbf{p} one obtains:

$$\Delta\mathbf{p} = t\nabla\phi(\mathbf{p}). \quad (2-9)$$

Then:

$$\phi(\mathbf{p} + \Delta\mathbf{p}) \approx \phi(\mathbf{p}) + \nabla\phi(\mathbf{p}) \cdot t\nabla\phi(\mathbf{p}) = 0 \quad (2-10)$$

$$t = \frac{\phi}{|\nabla\phi|^2} \quad (2-11)$$

$$\Delta\mathbf{p} = \frac{\phi}{|\nabla\phi|^2} \nabla\phi \quad (2-12)$$

The distance function for three examples is illustrated in Figure 16(c), Figure 17(c) and Figure 18(c). Boundary points are obtained by following the gradient from the interior point to the boundary. The resulting meshes are depicted in Figure 16(d), Figure 17(d) and Figure 18(d). Following this approach, valid quasi-structured meshes are obtained in most situations including problems with complicated boundaries as shown in Figure 17. However, the algorithm tends to produce a number of buffer layer elements with poor quality regarding the equiangle skewness and size density.

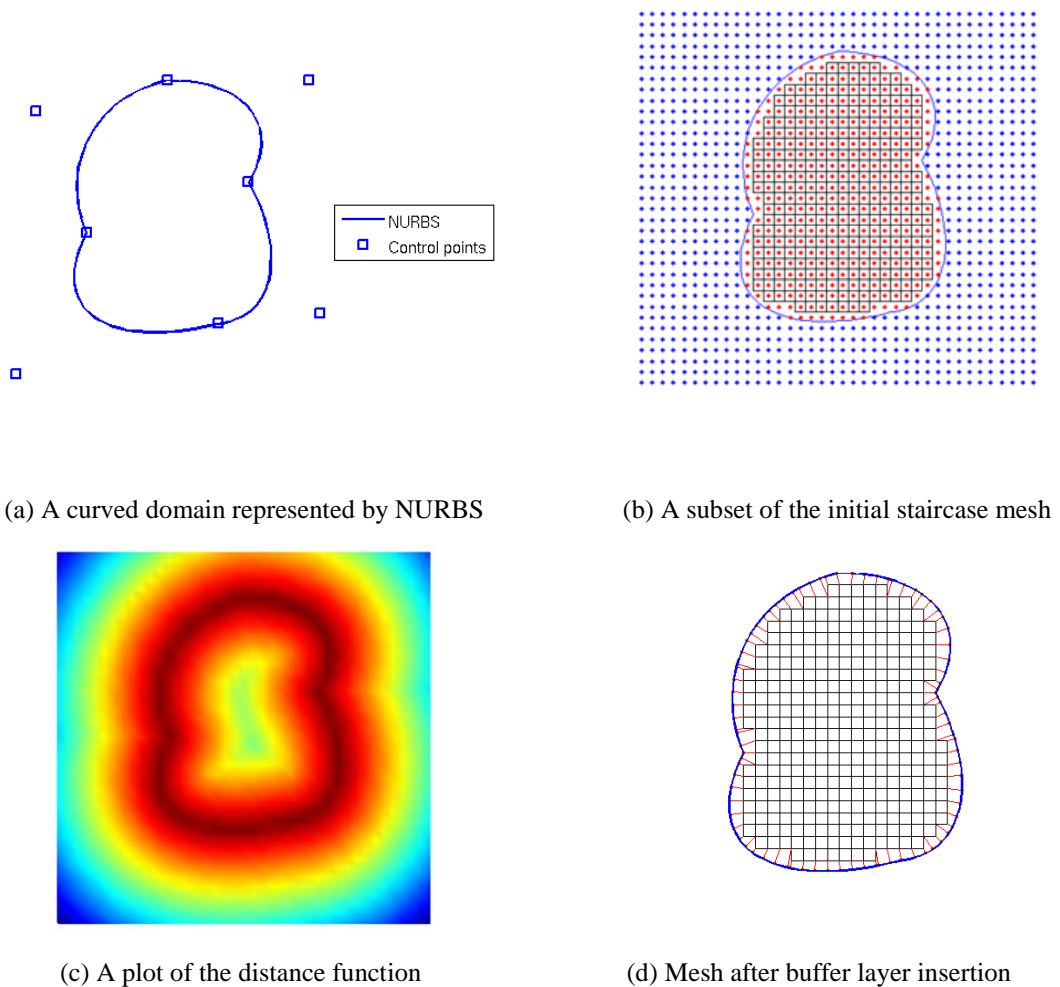
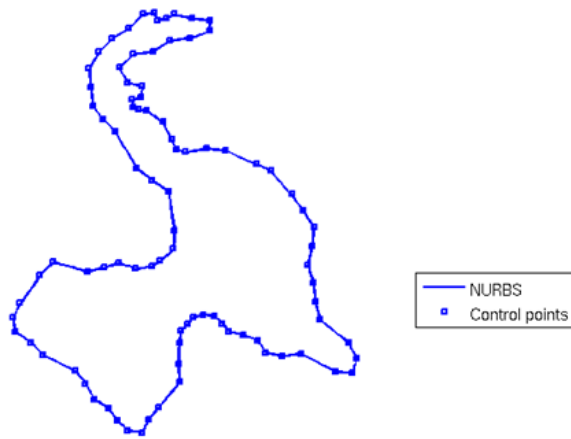


Figure 16 : Arbitrary curved boundaries can be described using NURBS (a). During the meshing process first an initial staircase mesh is generated (b), the distance functions is computed (c) and then the buffer layer elements are inserted between the boundary and the interior staircase mesh.

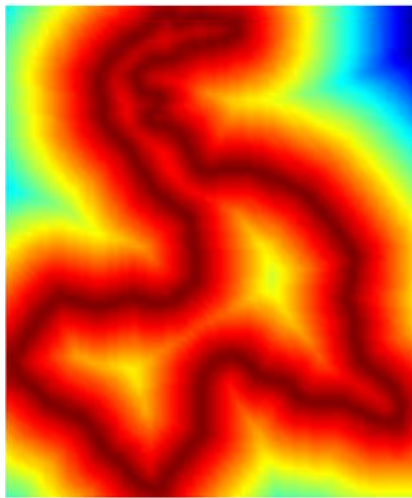
As an example in Figure 16, the boundary is implicitly described with a set of discrete points. In the second example in Figure 17 the boundary of the curved domain is constructed starting with the original implicit boundary, which then is parameterized by piecewise linear NURBS. The third example in Figure 18 is designed using piecewise quadratic NURBS.



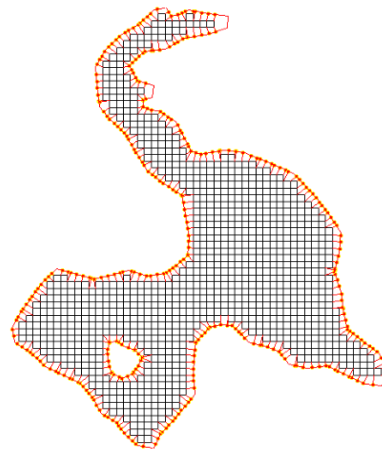
(a) The implicit boundary for a set of discrete points



(b) A set of initial staircase mesh

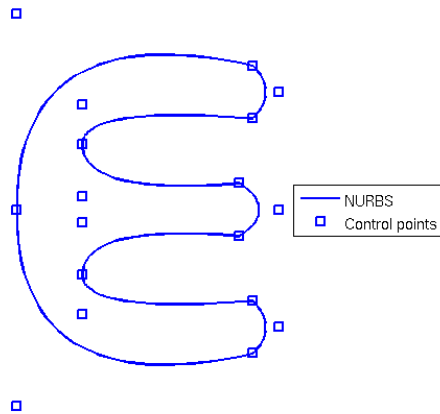


(c) A plot of distance function

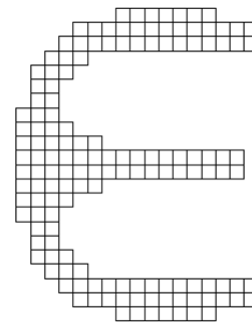


(d) The domain of lake is covered with buffer elements and structured mesh

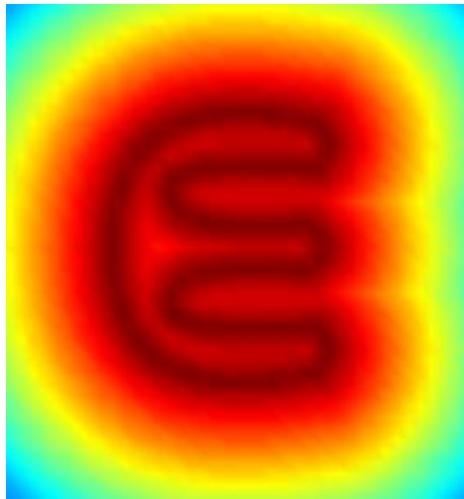
Figure 17 : Mesh example obtained by the buffer mesh scheme using NURBS, which employ piecewise linear basis functions.



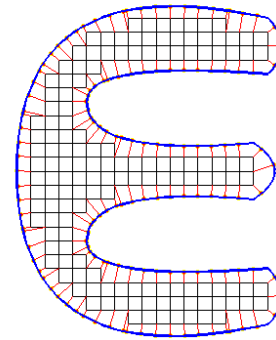
(a) The boundary of an “E-shaped” object described by piecewise quadratic NURBS



(b) A set of initial staircase mesh for “E-shaped” object



(c) A plot of distance function



(d) Final mesh with buffer elements and structured interior mesh

Figure 18 : Mesh example obtained by the buffer mesh scheme using NURBS, which employ piecewise quadratic basis functions.

2.3. High Order Scheme with Curved Elements

Up to now, the improvement of the representation quality of meshes has been achieved by increasing the number of elements for obtaining a finer mesh in the local sense, by shifting element vertices or by allowing the mesh to be unstructured to a certain degree. These techniques can be summarized as h - and r -refinement techniques (see Section 1.1.2). In these cases element edges have always been straight (aligned with Cartesian axes or linear in these coordinates). However, in the context of high order Finite Element and Discontinuous Galerkin Methods, representation quality of the mesh can be further

improved within the element, which is based on p - and r -refinement (see Section 1.1.2). In this situation, the concept of isoparametric elements [62] is invoked, where the same type of polynomial functions are applied for both the geometry and the solution approximations.

In order to realize a high order accurate geometry description, high order interpolations using polynomials are carried out. This leads to elements with non-straight edges, which are commonly called curved or curvilinear as they can be described with curvilinear coordinates by using locally invertible transformations [55]. From the implementation point of view, a high order interpolation within an element can be realized with a number of basis functions. Lagrange interpolating polynomials are the most common ones but other bases such as Legendre polynomials are possible too.

Another class of methods employing an isoparametric approach is the so-called isogeometric analysis [63, 64]. For such methods, NURBS basis functions are used for representing the boundary, as well as for approximating the unknown solutions. In this thesis, however, we remain with the more classical approach of a polynomial isoparametric method.

2.3.1 Curved Elements with the Transfinite Interpolation Technique

The buffer layer technique detailed in the preceding sections is very suitable for the application with curved elements as the elements to be curved are exactly those contained in the buffer layer. In order to turn straight-sided elements into high order curved elements their edges are deformed to the boundary by the Transfinite Interpolation (TFI) technique.

TFI is an algebraic mesh generation technique that is widely used in computational science. This does not only include the generation of meshes for Finite Element Methods but also computational design and imaging [72, 73]. It performs a mapping from a unit square (cube) to an area (volume) with almost arbitrary boundaries.

The advantage of TFI is that the interpolation procedure generates mesh points that are conforming to specific boundaries in the physical domain. Due to the algebraic characteristics, a TFI mesh is generated very fast, and its topology is equivalent to that of the initial structured mesh in the reference domain. The name Transfinite Interpolation is used to express that the resulting interpolating function can coincide with the exact function at infinitely many points.

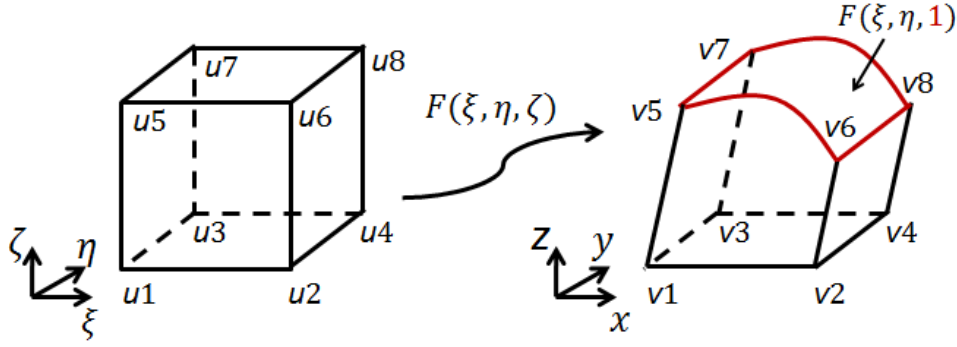


Figure 19 : A mapping function transforms a hexahedron from reference coordinates (ξ, η, ζ) to physical coordinates (x, y, z) with a curved surface $\{v5, v6, v7, v8\}$.

One issue related to the practical application of TFI is that the mapping from the reference to the physical domain is described by an analytic function, which can be very hard or impossible to find globally for complicated setups. One therefore resorts to performing TFI element-wise so that the deformation is relatively small within each element. Figure 19 describes the mapping from reference to physical coordinates.

The mapping function $F(\xi, \eta, \zeta)$ expresses the transformation of a reference hexahedron given in reference coordinates (ξ, η, ζ) to physical coordinates (x, y, z) . It is a vectorial function with each component expressing the transformation of a point in reference space to the value of one coordinate in physical space:

$$F(\xi, \eta, \zeta) = \begin{bmatrix} x(\xi, \eta, \zeta) \\ y(\xi, \eta, \zeta) \\ z(\xi, \eta, \zeta) \end{bmatrix} \quad (2-13)$$

From this transformation the Jacobi matrix can be computed, which plays an essential role for the transformation of derivatives with respect to either reference or physical coordinates from one to the other. The Jacobi matrix reads:

$$J(\xi, \eta, \zeta) = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}. \quad (2-14)$$

Its determinant is called the Jacobian, denoted by $|J(\xi, \eta, \zeta)|$. The value of the Jacobian is a measure of how the volume of an infinitesimal element of space changes during the transformation between coordinates. For simple affine transformations, such as size scaling, the Jacobian is a positive constant strictly greater than zero, which corresponds to the ratio of the sizes before and after the transformation. Degenerate elements described above and obtained by the mesh scheme used for generating the mesh shown in Figure

13(c) yield Jacobians with a negative or zero value at the position of degeneration. For non-affine transformations, the Jacobian is a function of the coordinates, meaning that it takes different values at different positions. A simple example of a non-affine transformation is the mapping of the reference square to a trapezoid, where the change in the size of any small area element depends on the position within the square.

Instead of applying the TFI technique to volumetric elements, it is typically applied for projecting element edges onto the curved surfaces of geometry. As the number of nodes placed immediately on the curved-edge can be infinite it is very suitable for numerical methods, which require higher order geometry approximation.

In order to describe any location in the interior of an element blending functions are employed. For a hexahedron, there are six linear blending functions $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$ for positive and negative directions in terms of ξ , η and ζ in the reference coordinates.

$$\begin{cases} \beta_1(\xi) = 1 - \xi \\ \beta_2(\xi) = \xi \\ \beta_3(\eta) = 1 - \eta \\ \beta_4(\eta) = \eta \\ \beta_5(\zeta) = 1 - \zeta \\ \beta_6(\zeta) = \zeta \end{cases} \quad (2-15)$$

Blending functions are a key issue of TFI as they control the smoothness of the element-local transformation.

For simplicity, we demonstrate the steps involved with TFI in Figure 20, where a quadrilateral is transformed from reference coordinates (ξ, η) to physical coordinates (x, y) with a curved edge $\{v_3 v_4\}$. This edge is parameterized by $(x, y) = F(\xi, 1)$. Figure 20(b) shows a set of interior points after the linear part of the mapping has been applied. For straight-sided elements, the mapping process is finished at this point. For curved elements however, the points with indices 2 to 5 first have to be made to coincide with the curved edge. In a last step, the blending functions are applied to generate a smooth distribution of all other nodes within the element.

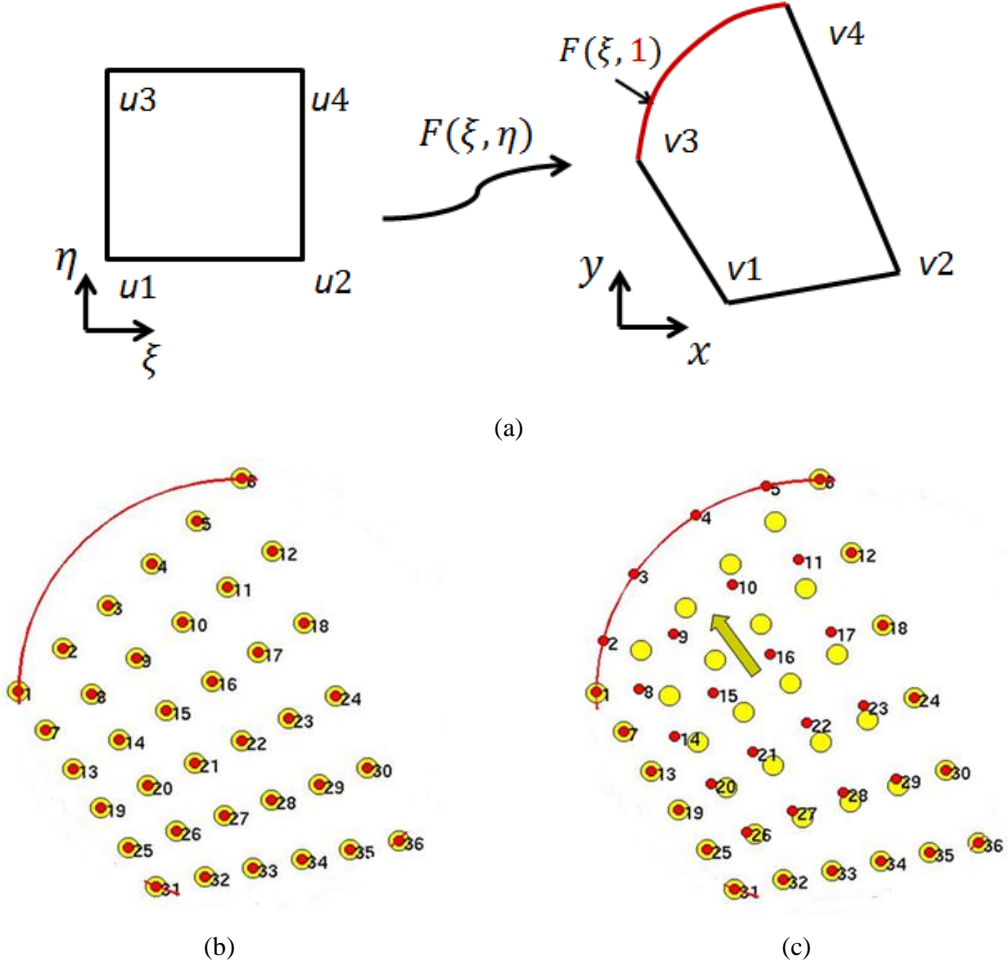


Figure 20 : A mapping function in transforms a quadrilateral from reference coordinates (ξ, η) to physical coordinates (x, y) with a curved edge $\{v3, v4\}$, which is parameterized by $(x, y) = F(\xi, 1)$ (a). A TFI implementation is demonstrated for a patch with uniform distributed points in (b) and (c).

The computations going along with these steps are the projection of the four parametric boundaries $F(\xi, 0)$, $F(\xi, 1)$, $F(0, \eta)$, $F(1, \eta)$. This is achieved by:

$$\begin{cases} \Pi_{\xi} \mathbf{F} = \beta_1 F(0, \eta) + \beta_2 F(1, \eta) \\ \Pi_{\eta} \mathbf{F} = \beta_3 F(\xi, 0) + \beta_4 F(\xi, 1) \end{cases} \quad (2-16)$$

$$\begin{cases} (\Pi_{\xi} \Pi_{\eta} \mathbf{F} = \beta_1 (\Pi_{\eta} \mathbf{F}(0, \eta)) + \beta_2 (\Pi_{\eta} \mathbf{F}(1, \eta))) \\ \Pi_{\xi} \Pi_{\xi} \mathbf{F} = \Pi_{\xi} \mathbf{F} \\ \Pi_{\eta} \Pi_{\eta} \mathbf{F} = \Pi_{\eta} \mathbf{F} \end{cases} \quad (2-17)$$

Then, the mapping within the complete element can be expressed as a Boolean sum according to the blending functions:

$$\mathbf{F}(\xi, \eta) = (\Pi_{\xi} \oplus \Pi_{\eta}) \mathbf{F} = (\Pi_{\xi} + \Pi_{\eta} - \Pi_{\xi} \Pi_{\eta}) \mathbf{F}, \quad (2-18)$$

$$\begin{aligned} \mathbf{F}(\xi, \eta) &= (\beta_1, \beta_2) \begin{pmatrix} \mathbf{F}(0, \eta) \\ \mathbf{F}(1, \eta) \end{pmatrix} + (\mathbf{F}(\xi, 0), \mathbf{F}(\xi, 1)) \begin{pmatrix} \beta_3 \\ \beta_4 \end{pmatrix}, \\ &- (\beta_1, \beta_2) \begin{pmatrix} \mathbf{F}(0, 0) & \mathbf{F}(0, 1) \\ \mathbf{F}(1, 0) & \mathbf{F}(1, 1) \end{pmatrix} \begin{pmatrix} \beta_3 \\ \beta_4 \end{pmatrix}. \end{aligned} \quad (2-19)$$

In combination with high order methods, TFI is an essential technique, in the sense that the high order accuracy of the field approximation is not degraded through low order geometry descriptions.

2.3.2 Numerical Integration

As will be detailed in the following chapter, a large number of integrals have to be evaluated when simulations with the Discontinuous Galerkin Method are to be performed. These integrals are not computed analytically but numerically using numerical integration techniques, i.e. quadrature rules [66]. The integrals to be computed can include both the Jacobi matrix and the Jacobians introduced above. In order to be confident with the correctness of the implementation of TFI and high order curved elements including all its intermediate steps, this section aims at investigating integration errors in area calculations of domains covered with meshes of curved elements.

A quadrature rule is a weighted sum of function values at specific points in the integration domain [66–69]. Given the number of quadrature points n in a one dimensional interval $[-1, 1]$, the Legendre–Gauss–Lobatto quadrature rule integrates polynomials up to order $2n - 3$ exactly. We call it Lobatto quadrature for short. We choose the Lobatto quadrature rule because its set of nodes always includes the interval end points, which is not the case for the classical Gauss quadrature. For a function $f(x)$ in the interval $[-1, 1]$, the general formula of the Lobatto quadrature is given by:

$$\int_{-1}^1 f(x) dx = \omega_1 f(-1) + \omega_n f(1) + \sum_{i=2}^{n-1} \omega_i f(x_i) \quad (2-20)$$

where ω_1 and ω_n are the weights for the fixed abscissas -1 and 1 :

$$\omega_1 = \omega_n = \frac{2}{n(n-1)}. \quad (2-21)$$

For $i = 2, \dots, n-1$, the free abscissas x_i are the roots of the polynomial $P'_{n-1}(x_i) = 0$, where $P(x)$ is the Legendre polynomial:

$$\omega_i = \frac{2}{n(n-1)[P_{n-1}(x_i)]^2}. \quad (2-22)$$

The tensor product of an interval $[-1, 1]$ leads to a square element $[-1, 1] \times [-1, 1]$ in

the reference domain. Given n quadrature points in each dimension, tensor products of one dimensional formulas extend the Lobatto quadrature into higher dimensions:

$$\int_{-1}^1 \int_{-1}^1 f(x, y) dx dy = \sum_{i=1}^n \sum_{j=1}^n \omega_i \omega_j f(x_i, y_j) \quad (2-23)$$

Lobatto points described in two dimensions include the points on the edges of elements.

In order to validate the mesh scheme, the area and circumference of a circular domain is computed. The scheme is applied to generate meshes using different orders of the geometry representation ranging from linear (one) to six. The difference between numerical and analytical results implies the geometric approximation errors. Under mesh refinement, i.e., reduction of the element size, the error in the area and circumference should decrease at a rate given by the representation order. Choosing a circular domain has the advantage that it has no exact explicit representation using polynomials. Therefore, each order increase should lead to a faster decay rate of the approximation error.

Figure 14(a) shows an exemplary discretization of the circular domain using a buffer layer mesh. In the visualization the TFI procedure is carried out for the curved elements with five-by-five uniformly distributed interpolation points. The same interpolation routine is employed to construct the quadrature nodes.

For the circumference and the area of the circle, we perform numerical integration with edge points and area points respectively. The edge integration employs one dimensional Lobatto quadrature, while the volume integration is applied with the tensor product quadrature we have just introduced. The quadrature order, i.e., the number of quadrature nodes, has to be large enough to give an accurate result taking into account that the Jacobian is a non-linear function of the coordinates as described above. The integration operation is carried out element-wise and the individual results are added up for obtaining the total area and circumference of this unit circle.

Figure 21 depicts the numerical integration errors with respect to the number of elements. As the number of elements increases, i.e., the element sizes get smaller, all the errors decrease. As expected the integration order determines the rates at which the errors decrease. With the order increasing from 1st to 6th, the convergence rate increases accordingly. For 6th order elements, the smallest error reaches the floor of double precision and no further reduction can be achieved. This geometric integration example shows the accuracy of the mesh scheme in terms of integration error convergence orders. It shows that the construction of high order curved elements is correct from the point of view of derivation as well as implementation.

Although high order quadrature is performed for each element in the mesh, only the curved ones benefit from the high order. For the straight-sided elements, first order quadrature is sufficient to achieve an exact result. Consequently, high order integration has to be applied for high order curved elements only.

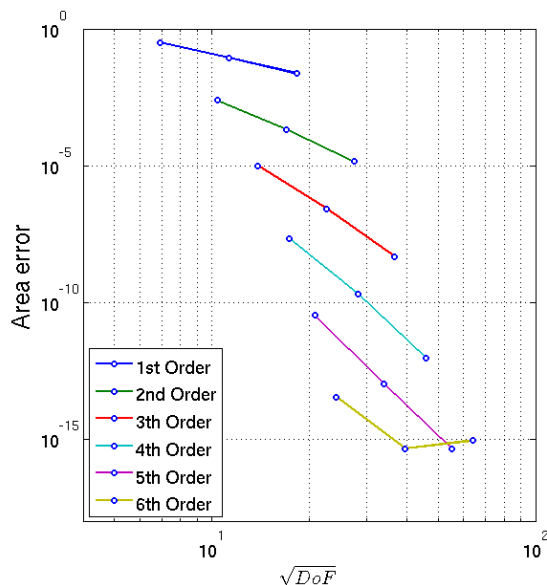
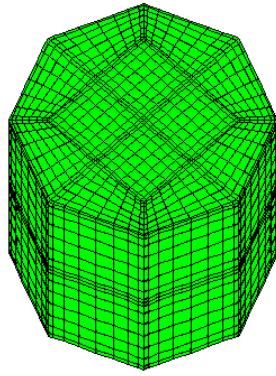


Figure 21 : Numerical integration test of a circular domain area using the high order curved element mesh scheme with different quadrature orders.

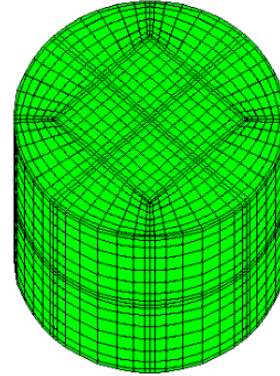
2.4. Extending the Mesh Scheme into Three Dimensions

The generation of meshes in three-dimensional space is known to be significantly more difficult than in lower dimensions. However, in comparison to a boundary-first scheme, the extension of the proposed mesh-first scheme into three-dimensional space can be achieved more easily. The principle procedure remains unaltered. First, a background mesh with hexahedral elements is built. Then a buffer region is left between the boundary and the interior structured mesh. The elements in the buffer layer are then constructed using the distance function approach presented above. As a consequence, the mesh scheme succeeds in constructing meshes that are quasi-structured (fully structured within the interior mesh) also in three-dimensional applications. Other advanced topics such as embedding small geometry features can be found in [21].

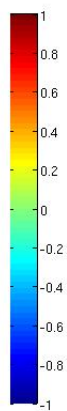
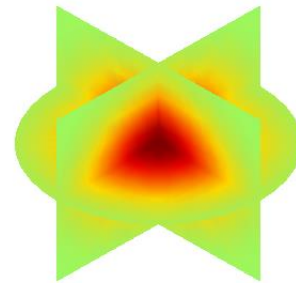
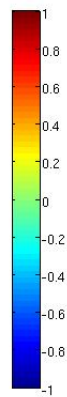
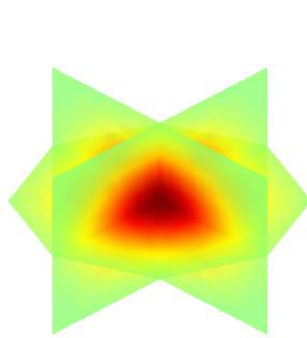
In Figure 22, we compare the mesh resolutions between curved elements and straight elements in cylinder geometry. The surfaces of elements in plot (a) are piecewise plain, and those in plot (b) are fitting to the curved boundary under r -refinement.



(a) A polyhedral mesh of a cylinder using 24 hexahedral elements with only plain surfaces



(b) An exact mesh of a cylinder using 24 hexahedral elements with curved surfaces



(c) A cut view of the field distribution using the polygon mesh for the cylinder.

(d) A cut view of the field distribution using an exact mesh of the cylinder.

Figure 22 : Three-dimensional Gaussian distributions of electric field in (a) polyhedral mesh and in (b) polyhedral domains truncated from the sphere.

By setting Gaussian distributions of electric fields, it's not surprise to observe volume missing in plot (c), since its corresponding mesh in (a) can be seen as a truncation of the cylinder domain. On the contrary, the complete field distribution in the domain is represented in (d) based on mesh in (b), which is obtained by TFI technique as discussed in Section 2.3.

As an example, we apply the buffer mesh scheme to a sphere. Figure 23 depicts the construction process for two different mesh resolutions. The structured interior mesh is depicted on the right of (a) and (b) along with the faces whose vertices have to be projected onto the surface for obtaining the buffer layer elements. In (c) and (d) the final meshes are shown. As can be seen from the illustration, the outer elements are straight-sided. The curved elements can be obtained by performing the TFI routines examined in Figure 22 from the last example.

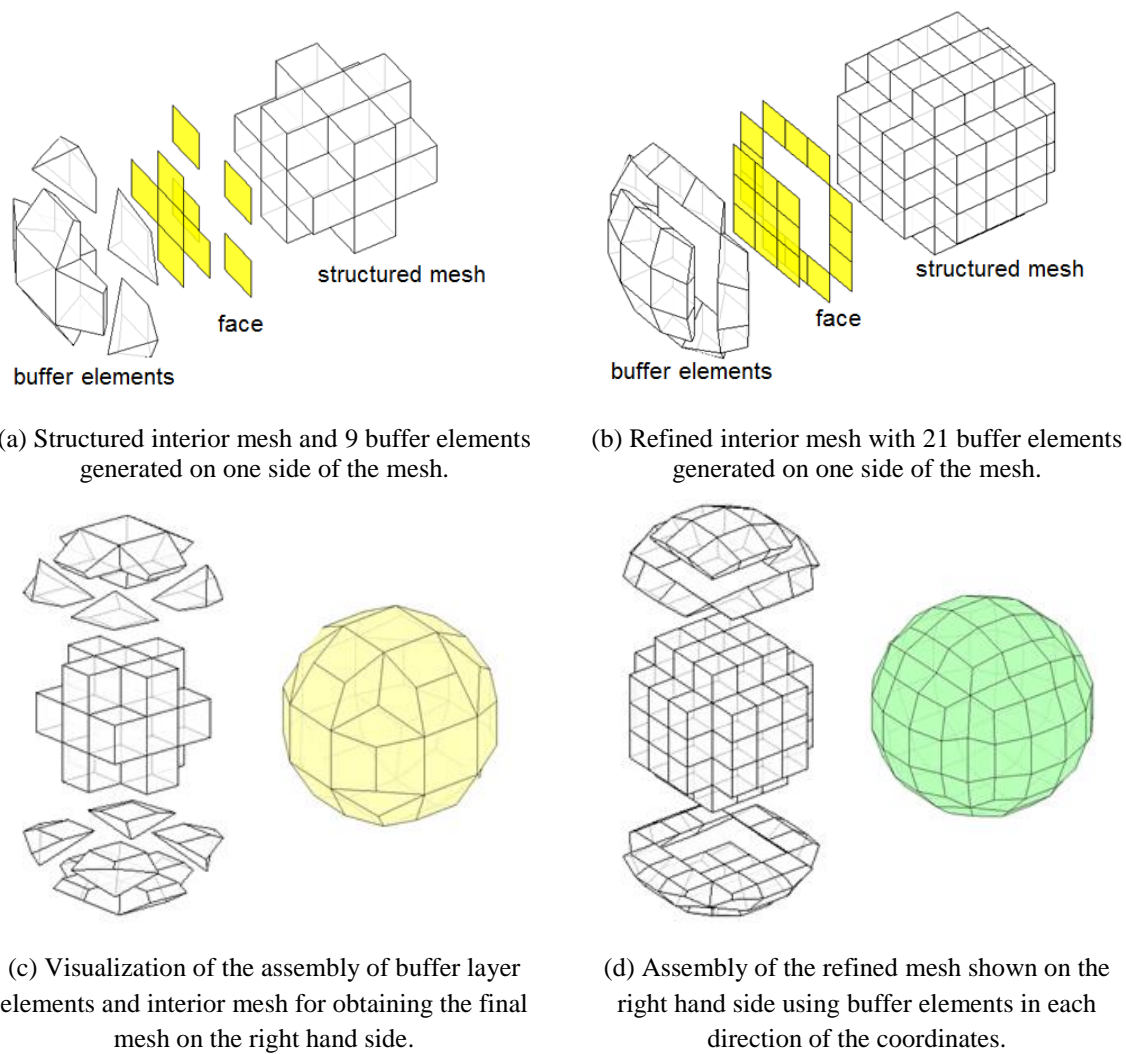


Figure 23 : Examples of the generation of body-fitting meshes based on a structured interior mesh and its corresponding buffer layers elements in three dimensions.

In Figure 24, the electric field is initialized with a Gaussian distribution in a sphere shape domain. Field resolutions on the spherical surface are improved from (c) (f) with plain face elements to (b) (e) with curved elements. Based on the sphere with curved elements, we perform the three dimensional numerical integrations, which is similar to the two dimensional routines in Section 2.3. The high order convergence rates in terms of volume errors have been demonstrated by Figure 25 with different quadrature orders.

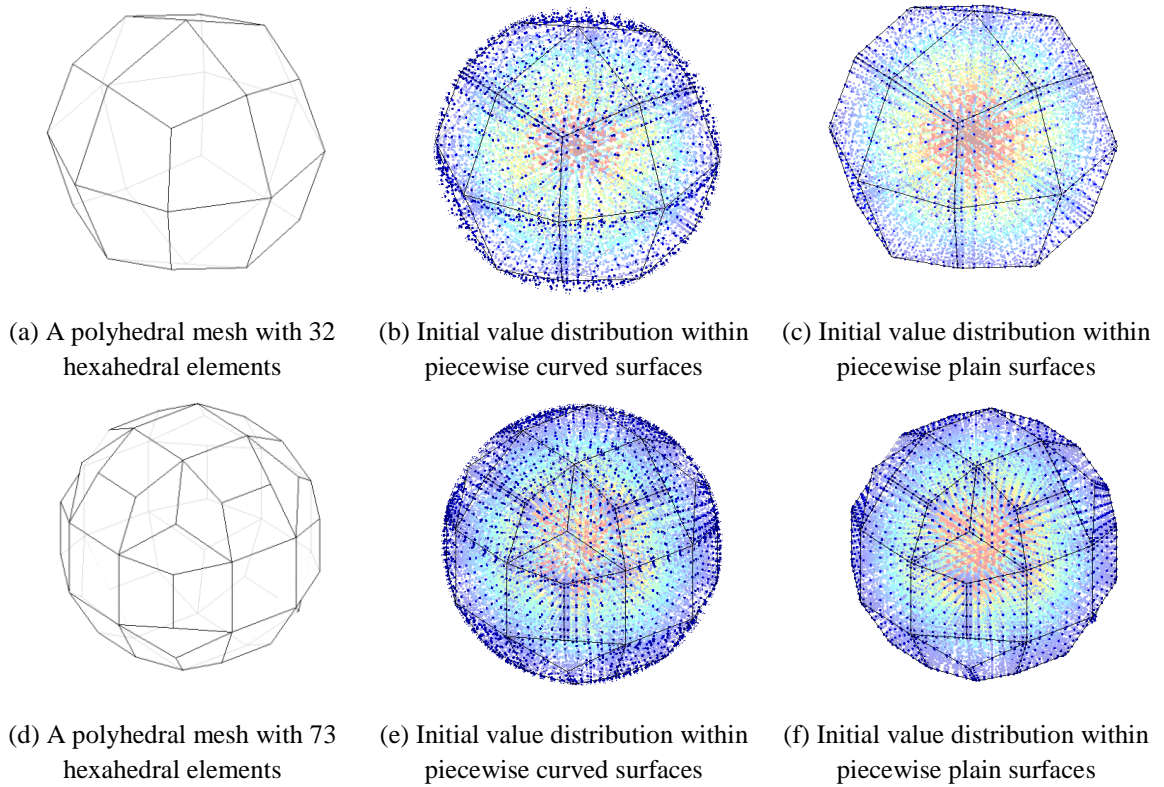


Figure 24 : Three-dimensional Gaussian distributions of electric field in (b) (e) spherical domains and in (c) (f) polyhedral domains truncated from the sphere.

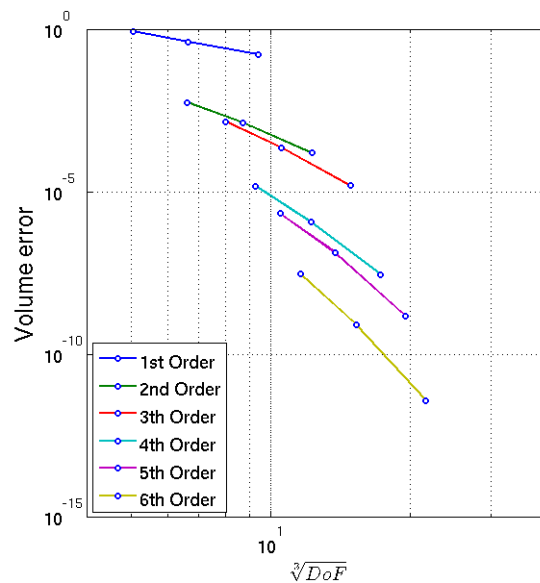
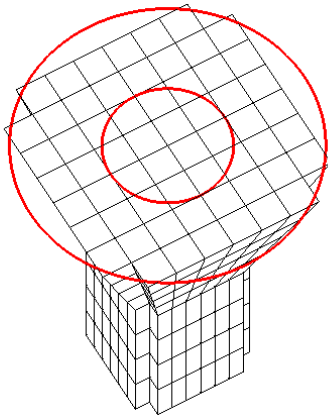
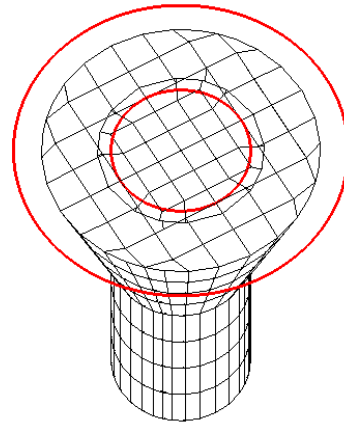


Figure 25 : Numerical integration test of a sphere domain volume using the high order curved element mesh scheme with different quadrature orders.

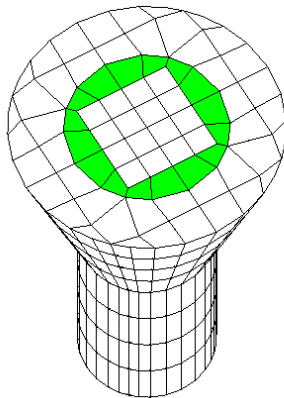
As another three-dimensional example, the generation of a mesh for a cylindrical horn antenna is presented in Figure 26. The walls of the antenna (marked with red lines) have been artificially thickened for demonstration purposes. The mesh construction works entirely in the same way for real antennas. The first step is again the extraction of the structured interior mesh from the background mesh as shown in (a). In (b) it is demonstrated again that without the application of a buffer layer degenerated elements occur when the mesh is fitted to rounded surfaces. On the antenna inside, the fitting process makes use of the proposed buffer layer technique (c). The interior mesh of the horn remains fully structured. In (d) finally, a buffer layer is inserted at the outer surface of the horn as well, yielding the final mesh.



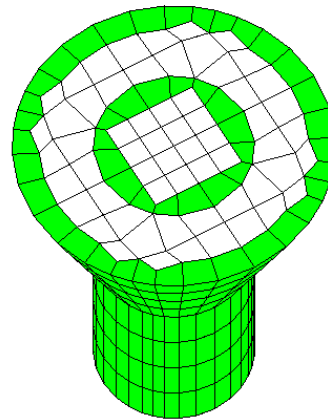
(a) Staircase elements in the domain



(b) Deformed elements near the boundary



(c) Inner buffer layer with inserted elements



(d) Outer buffer elements with inserted elements

Figure 26 : Meshing a circular horn antenna. Elements generated in the buffer layers are geometry-oriented, i.e. new elements (in green color) are inserted along the surfaces of the geometry.

3. Discontinuous Galerkin Method

The scope of this section is on introducing and describing the Discontinuous Galerkin Method (DGM). As a high order numerical method, the DGM has the potential ability to achieve the best convergence behavior possible, i.e., exponential convergence. This chapter will also fill the gaps left intentionally in Chapter 2 for the sake of separating the mesh scheme from the numerical method. The DGM is a Finite Element-type method, and we will follow the definition of Finite Elements and investigate the ingredients required for high order methods. Two sets of basis functions, namely Lagrange and Legendre polynomials, will be compared, and the advantages of utilizing the latter for our purposes will be described. After a review of Maxwell's equations, we proceed to describe their discretization with a DGM formulation.

3.1. High Order Finite Element Methods

As a formal definition given by Ciarlet [56], a finite element is a triple (K, P, Σ) where:

- (1) K is a closed domain with piecewise smooth boundary in \mathbb{R}^d .
- (2) P is a finite space of functions with N_p dimensions defined over domain K .
- (3) Σ is a set of N_p linear forms $\{L_1, L_2, \dots, L_{N_p}\}$. The elements of Σ are called the Degrees of Freedom (DoF).

The geometric element domain K is commonly referred to as interval ($d = 1$), triangle or quadrilateral ($d = 2$), and tetrahedron or hexahedron ($d = 3$). A set of functions $\mathcal{B} = \{\theta_1, \theta_2, \dots, \theta_{N_p}\} \subset P$ is said to be a set of unique basis functions, if they fulfill the δ -property:

$$L_i(\theta_j) = \delta_{ij} \quad \text{for } 1 \leq i, j \leq N_p. \quad (3-1)$$

Here δ_{ij} is the standard Kronecker delta, $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise.

The aim is to choose a set of proper basis functions from the finite element space P . We compare two sorts of bases: Lagrange and Legendre polynomials and check whether their

corresponding linear forms satisfy the δ -property. In Figure 27, both polynomials with 7th order are presented in the reference domain $[-1,1]$.

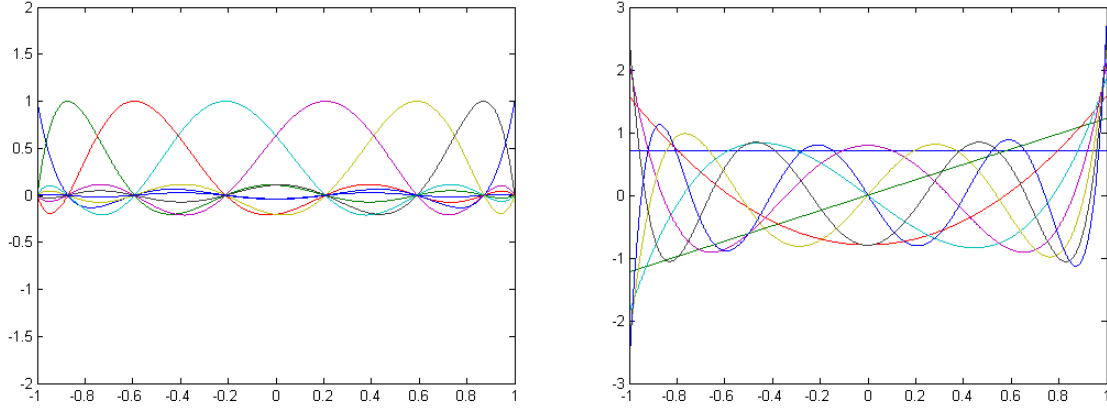


Figure 27 Lagrange polynomial basis functions of order 7 (on the left hand side). Legendre polynomial basis functions of order 7 (on the left hand side).

The family of Lagrange polynomials is studied [56]. Assume (K, P, Σ) is a finite element. The linear forms $\{L_1, L_2, \dots, L_{N_p}\}$ are defined as sampling function values at the interpolation points. If there exists a set of points $\{\xi_1, \xi_2, \dots, \xi_{N_p}\}$ in the element domain K , such that, for all functions $g \in P$:

$$L_i(g) = g(\xi_i) \quad \text{for } 1 \leq i \leq N_p, \quad (3-2)$$

then (K, P, Σ) is called a Lagrange finite element. We check the linear forms with Lagrange basis functions $\mathcal{B} = \{\phi_1, \phi_2, \dots, \phi_{N_p}\}$ where

$$\phi_i(\xi_j) = \delta_{ij} = \begin{cases} 0 & (i \neq j) \\ 1 & (i = j) \end{cases} \quad \text{for } 1 \leq i, j \leq N_p, \quad (3-3)$$

By plugging function ϕ_j into the linear form (3-2), it is verified that the δ -property is fulfilled:

$$L_j(\phi_i) = \phi_i(\xi_j) = \delta_{ij}. \quad (3-4)$$

Following the same routine, we examine a finite element with Legendre basis functions. Let (K, P, Σ) be a finite element. The linear forms $\{L_1, L_2, \dots, L_{N_p}\}$ are defined as a particular inner product. Assume a set of functions $\mathcal{B} = \{\varphi_1, \varphi_2, \dots, \varphi_{N_p}\}$ in the element domain K , such that, for all functions $g \in P$:

$$L_i(g) = \frac{1}{2} \int_{-1}^1 \varphi^i(\xi) g(\xi) d\xi \quad \text{for } 1 \leq i \leq N_p. \quad (3-5)$$

Using scaled Legendre polynomials [74], the basis functions \mathcal{B} have the property of being pairwise orthogonal within the interval $[-1,1]$:

$$\frac{1}{2} \int_{-1}^1 \varphi^i(\xi) \varphi^j(\xi) d\xi = \delta_{ij} = \begin{cases} 0 & (i \neq j) \\ 1 & (i = j) \end{cases} \text{ for } 1 \leq i, j \leq N_p. \quad (3-6)$$

By equipping function φ^j with the linear form equation (3-5), the δ -property is fulfilled:

$$L_i(\varphi^j) = \frac{1}{2} \int_{-1}^1 \varphi^i(\xi) \varphi^j(\xi) d\xi = \delta_{ij}. \quad (3-7)$$

In the Lagrange finite element, the basis functions interpolate function values for each of the points $\{\xi_1, \xi_2, \dots, \xi_{N_p}\}$, as shown in equation (3-2). Due to their interpolation property in equation (3-3), Lagrange polynomials are often used as nodal basis. However, such basis functions are constructed for a specific order. They have to be recomputed when the order of the element is altered.

Figure 28 shows Lagrange polynomials with the Legendre–Gauss–Lobatto interpolation points, which have the same distribution as the Lobatto quadrature points explained previously in Section 2.3.2. When increasing the order, we obtain three different sets of basis functions. The basis functions of lower order $\mathcal{B}^p = \{\varphi_1, \varphi_2, \dots, \varphi_p\}$ cannot be reused in the high order case $\mathcal{B}^{p+1} = \{\varphi_1, \varphi_2, \dots, \varphi_{p+1}\}$.

It is worth concerning about the efficiency of the scheme when constructing a high order method. An alternative approach are the so-called hierarchical basis polynomials. In a hierarchical basis, the basis is extended to a higher order \mathcal{B}^{p+1} by adding an additional higher order basis function such that $\mathcal{B}^p \cup \{\varphi_{p+1}\} = \mathcal{B}^{p+1}$. As a result, the expansion of a certain physical quantity is refined by adding additional modes to the existing basis. They are commonly referred to as modal bases.

In Figure 29 the hierarchical modal Legendre basis polynomials are illustrated. All lower order basis functions are included in the higher order ones. Modal bases do not satisfy the Lagrange interpolation property of equation (3-2) but they show the property of orthogonality in the L^2 -sense in the reference element. This property yields block diagonal matrix operators, which are largely advantageous in the numerical scheme with explicit time-stepping.

Nodal and modal basis functions are two forms to approximate a physical quantity. They can be transformed into each other conveniently through a Vandermonde matrix.

$$u(\xi) = \sum_{n=1}^{N_p} \hat{u}_n \varphi_n(\xi) = \sum_{i=1}^{N_p} u_i \phi_i(\xi), \quad (3-8)$$

with Legendre polynomial basis functions φ_n of order $n \in \{1, 2, \dots, N_p\}$, and Lagrange polynomial basis functions ϕ_i of order $i \in \{1, 2, \dots, N_p\}$.

Assemble the column vector $\varphi_n(\xi)$, the Vandermonde matrix V with entries $V_{i,n}$ given by:

$$V = [\varphi_1(\xi), \varphi_2(\xi), \dots, \varphi_{N_p}(\xi)], \quad (3-9)$$

$$V_{i,n} = \varphi_n(\xi_i). \quad (3-10)$$

According to the interpolation property of Lagrange polynomial in equation (3-3), the nodal coefficient u_i is written:

$$u_i = u(\xi) \phi_i(\xi) = u(\xi_i). \quad (3-11)$$

Then the nodal coefficients u_i are calculated with the modal coefficients by taking the modal expression of equations (3-8) (3-10) into equation (3-11) :

$$u_i = \sum_{n=1}^{N_p} V_{i,n} \cdot \hat{u}_n. \quad (3-12)$$

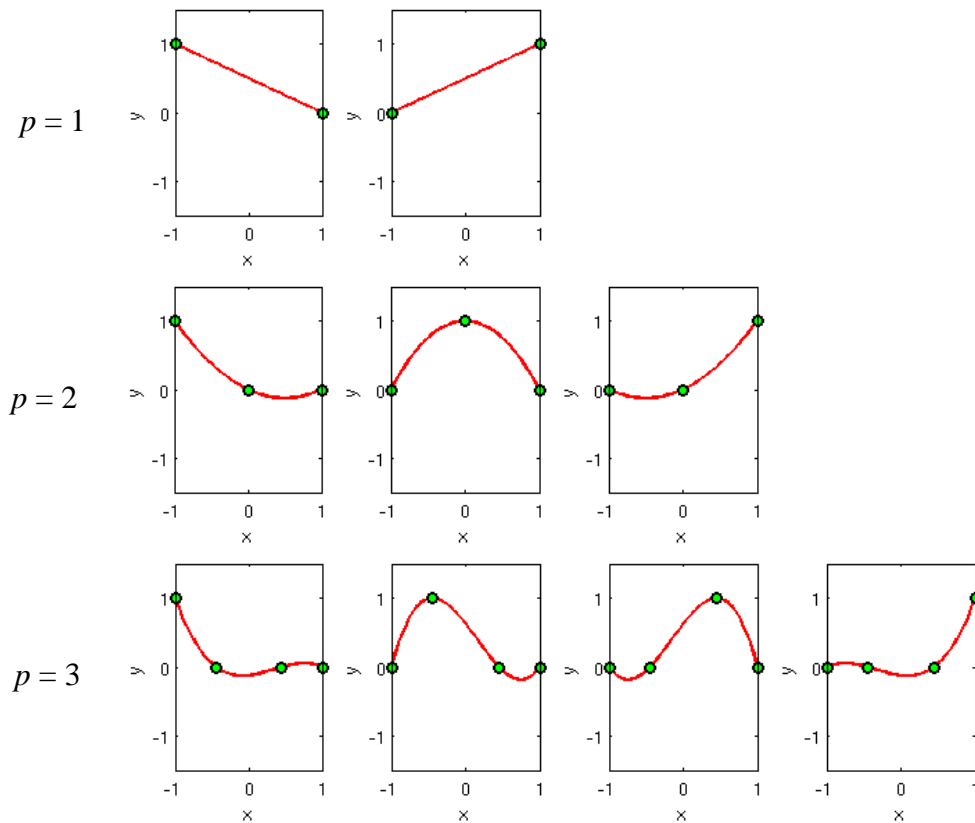
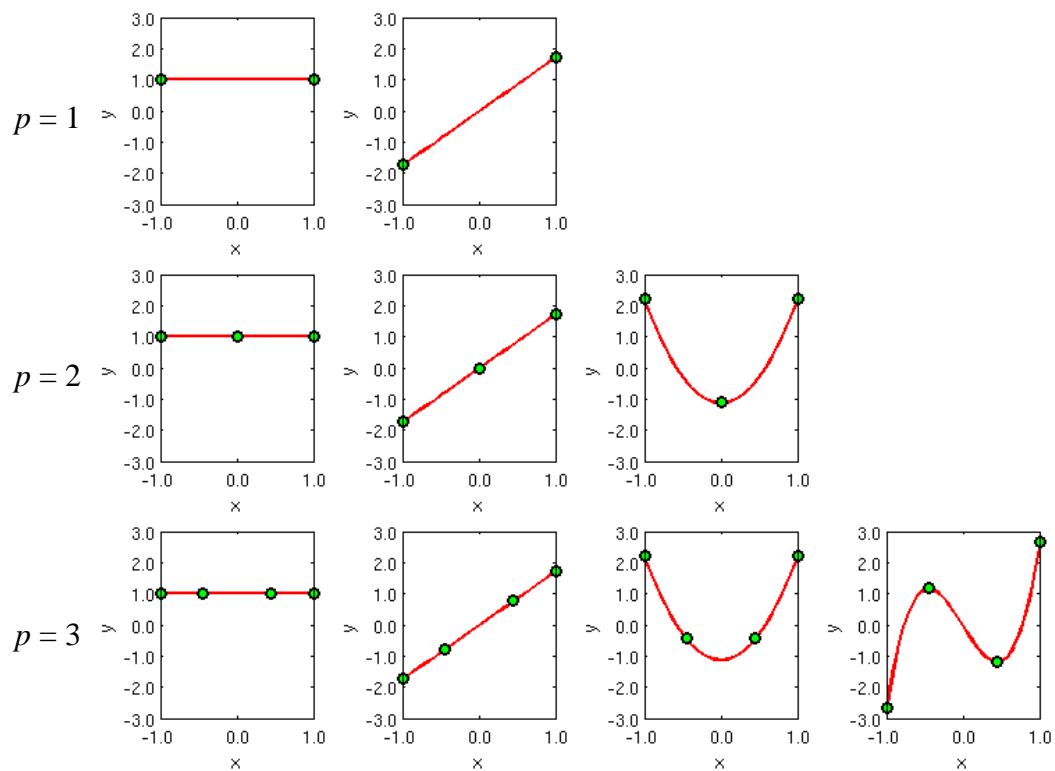
Assembling the vector $(u_1, u_2, \dots, u_{N_p})^T$, and taking the expression of u_i from equation (3-12), we have the transformation from modal coefficients to nodal coefficients:

$$(u_1, u_2, \dots, u_{N_p})^T = V (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{N_p})^T. \quad (3-13)$$

Given that the Vandermonde matrix V is invertible, the transformation back to modal coefficients is computed by:

$$(\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{N_p})^T = V^{-1} (u_1, u_2, \dots, u_{N_p})^T. \quad (3-14)$$

For the nodal-to-modal conversion in equation (3-14), the nodal points in the reference coordinates should be carefully chosen, such that V^{-1} is well-conditioned. As the polynomial order increases, Legendre–Gauss–Lobatto points yield smaller condition numbers than uniform points (see [8]).

Figure 28: Nodal basis functions with Lagrange polynomials from 1st to 3rd order.Figure 29: Hierarchical modal basis with Legendre polynomials up to 3rd order, where the lower order polynomials are all included in the higher order ones.

3.2. The Discontinuous Galerkin Method and its Properties

The Discontinuous Galerkin Method [8, 9, 10, 11] presents a numerical method, which has features of both the Finite Element Method [13, 14] and the Finite Volume Method [15 16]. The DGM has been used in a wide variety of applications in the last decades but in particular for time-dependent problems. Intensive studies can be found for the solution of hyperbolic systems describing wave propagation [11 18], such as computational electromagnetics, fluid dynamics and aeroacoustics.

As a finite element method, the DGM employs a set of basis functions for approximating the unknown solution in terms of an expansion with respect to this basis. Typically, polynomials are employed as basis functions, which can, in theory, be of arbitrarily high order p . The usual measure for describing the approximation quality in an asymptotic sense is the convergence order, which is the order at which the approximation error tends towards zero. It has been revealed that the DGM with element size h , converges at a rate $h^{p+1/2}$ in general and h^{p+1} in some cases [9, 11]. This is an attractive feature because the error will converge exponentially if the order p is increased, at least in many cases.

The most significant difference between the DGM and more traditional FEM approaches employing nodal [13] or edge elements [67 68] is the support of the basis functions, which is strictly local to one element in the DGM. As an immediate consequence, DG approximations are globally discontinuous, where discontinuities can occur at all interfaces between neighboring elements.

The discontinuous nature of the approximations shows the close relationship of the DGM to the Finite Volume Method. In fact, the DGM could also be understood as a generalized Finite Volume Method. As the element-wise approximations are complete uncoupled at first, the DGM lends the concept of coupling solutions between elements through a so-called numerical flux. The choice of the flux is essential for the stability, convergence and efficiency of the DGM [10, 15]. The numerical flux will be discussed in more details in Section 3.4.2, where the DG form of Maxwell's equations is presented.

3.3. Maxwell's Equations

Maxwell's equations are the fundamental equations for describing macroscopic electromagnetic phenomena. Their integral form in continuous space can be expressed as follows:

$$\int_{\partial S} \mathbf{H} \cdot d\mathbf{l} = \int_S \left(\frac{\partial \mathbf{D}}{\partial t} + \mathbf{J} \right) \cdot d\mathbf{S}, \quad (3-15)$$

$$\int_{\partial S} \mathbf{E} \cdot d\mathbf{l} = - \int_S \frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{S}, \quad (3-16)$$

$$\int_{\partial V} \mathbf{D} \cdot d\mathbf{S} = \int_V \rho \, dV, \quad (3-17)$$

$$\int_{\partial V} \mathbf{B} \cdot d\mathbf{S} = 0, \quad (3-18)$$

for any surface S , volume V and their respective boundaries ∂S and ∂V . Throughout this thesis, boundaries in general mean ending vertices of a line in one dimension, the closed outline of an area in two dimensions and the closed surface of a volume in three dimensions. The spatial variable is denoted by the vector \mathbf{r} , and the temporal variable by t . The vector quantities in equations (3-15), (3-16), (3-17) and (3-18) are:

$\mathbf{E}(\mathbf{r}, t)$ the electric field strength (Volt/meter),

$\mathbf{H}(\mathbf{r}, t)$ the magnetic field strength (Ampere/meter),

$\mathbf{D}(\mathbf{r}, t)$ the electric flux density (Ampere · second/meter²),

$\mathbf{B}(\mathbf{r}, t)$ the magnetic flux density (Volt · second/meter²),

$\mathbf{J}(\mathbf{r}, t)$ the electric current density (Ampere/meter²),

$\rho(\mathbf{r}, t)$ the electric charge density (Coulomb/meter³).

Equation (3-15) and equation (3-16) are referred to as Ampere's law and Faraday's law, respectively. The third equation (3-17) is named Gauss' law and the fourth equation (3-18) is named Gauss' law of magnetism which states the absence of magnetic charges. The current density $\mathbf{J}(\mathbf{r}, t)$ includes the conduction current density as well as externally imposed source currents.

Maxwell's equations in integral form are mathematically equivalent to their differential forms, which are derived by applying the Kelvin-Stokes theorem to equations (3-15) and (3-16), and the Gauss divergence theorem to equations (3-17) and (3-18). Consequently, the differential form of time-dependent Maxwell's equations can be written as follows:

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}, \quad (3-19)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (3-20)$$

$$\nabla \cdot \mathbf{D} = \rho, \quad (3-21)$$

$$\nabla \cdot \mathbf{B} = 0. \quad (3-22)$$

The charge conservation law is achieved by first multiplying the divergence operator on both sides of Ampere's law (3-19), and then inserting the resulting equation with Gauss' law (3-21):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0. \quad (3-23)$$

Additional constitutive equations are required for making the system solvable. They include the material characteristics of the underlying media. The symbols of the material properties $\varepsilon(\mathbf{r})$, $\mu(\mathbf{r})$ and $\kappa(\mathbf{r})$ denote permittivity, permeability and conductivity, respectively, and couple the flux (current) density and the field strength via the relations:

$$\mathbf{D}(\mathbf{r}, t) = \varepsilon(\mathbf{r})\mathbf{E}(\mathbf{r}, t), \quad (3-24)$$

$$\mathbf{B}(\mathbf{r}, t) = \mu(\mathbf{r})\mathbf{H}(\mathbf{r}, t), \quad (3-25)$$

$$\mathbf{J}(\mathbf{r}, t) = \kappa(\mathbf{r})\mathbf{E}(\mathbf{r}, t). \quad (3-26)$$

If $\varepsilon(\mathbf{r})$, $\mu(\mathbf{r})$ and $\kappa(\mathbf{r})$ do not depend on the value, direction and position of the electromagnetic field, the medium is called linear, isotropic and homogeneous, respectively. Another characteristic property is medium dispersion, which indicates that $\varepsilon(\mathbf{r})$, $\mu(\mathbf{r})$ and $\kappa(\mathbf{r})$ are frequency dependent. Throughout this thesis, the materials are assumed to be linear, isotropic, homogeneous and non-dispersive.

3.4. Discretization of the Maxwell's Equations

3.4.1 Governing Equations

Neglecting electric charge ρ and current source \mathbf{J} , the Maxwell's equations are written as follows:

$$\boldsymbol{\varepsilon} \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{H} = 0, \quad (3-27)$$

$$\boldsymbol{\mu} \frac{\partial \mathbf{H}}{\partial t} + \nabla \times \mathbf{E} = 0. \quad (3-28)$$

New symbols are introduced, including a material parameter $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\varepsilon} & 0 \\ 0 & \boldsymbol{\mu} \end{bmatrix}$, a vector $\mathbf{U} = \begin{bmatrix} \mathbf{E} \\ \mathbf{H} \end{bmatrix}$ and a linear form $\mathbf{F}(\mathbf{U})$, such that the system of equations (3-27) (3-28) can be rewritten in a conservative form:

$$\boldsymbol{\alpha} \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0. \quad (3-29)$$

The function $\mathbf{F}(\mathbf{U}) = \begin{pmatrix} 0 & -\boldsymbol{\beta} \\ \boldsymbol{\beta} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix}$ contains a matrix operator $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$, where $\beta_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, $\beta_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$ and $\beta_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. It can be verified that the divergence of $\mathbf{F}(\mathbf{U})$ in the conservative form is equivalent to the curl form:

$$\nabla \cdot \mathbf{F}(\mathbf{U}) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \nabla \times \mathbf{E} \\ \nabla \times \mathbf{H} \end{pmatrix}. \quad (3-30)$$

In a Cartesian frame, divergence and curl operators are defined as $\nabla \cdot = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$ and $\nabla \times = \left(\frac{\partial}{\partial x} \beta_1 + \frac{\partial}{\partial y} \beta_2 + \frac{\partial}{\partial z} \beta_3 \right)$ respectively. With each electromagnetic component cast into $\mathbf{U} = (E_x, E_y, E_z, H_x, H_y, H_z)^T$, the linear function $\mathbf{F}(\mathbf{U})$ is explicitly formulated as:

$$\mathbf{F}(\mathbf{U}) = \left[\begin{pmatrix} 0 \\ H_z \\ -H_y \\ 0 \\ -E_z \\ E_y \end{pmatrix}, \begin{pmatrix} -H_z \\ 0 \\ H_x \\ E_z \\ 0 \\ -E_x \end{pmatrix}, \begin{pmatrix} H_y \\ -H_x \\ 0 \\ -E_y \\ E_x \\ 0 \end{pmatrix} \right]. \quad (3-31)$$

A very common boundary condition is the Perfectly Electrically Conducting (PEC) boundary condition, which models the interface with an infinitely good conductor. It is expressed as:

$$\mathbf{n} \times \mathbf{E} = 0, \quad \mathbf{n} \cdot \mu \mathbf{H} = \mathbf{n} \cdot \mathbf{H} = 0, \quad (3-32)$$

where \mathbf{n} is the outward pointing normal vector and the above restrictions on the media apply. PEC boundary conditions are often used for cavity and scattering problems.

In the next section, the discretization of the conservative form of Maxwell's equations following the DG procedure is introduced.

3.4.2 Semidiscrete Formulation

The semidiscrete formulation of Maxwell's equations is obtained by discretizing the spatial dimensions while keeping the temporal dimension continuous. Let Ω_h denote a discretization of a continuous domain of interest Ω . Mesh generators are typically able to generate regular meshes, which do not contain curved edges, curved surfaces or hanging nodes resulting in a piece-wise linear approximation of the domain Ω by regular meshes. However, in this thesis a formulation applying discretizations with high order curved elements, as introduced in the preceding chapter is developed.

We reconsider the simple two-dimensional situations shown in Figure 30 where the continuous domain Ω consists of one quarter of a circle. The geometry is discretized by three mesh schemes, and the resulting representations are shown in green. As described previously an approximation using a purely Cartesian mesh as in Figure 30(a) can result in a poor representation, which is commonly referred to as the staircase approximation. A more accurate approach would be employing quadrilateral elements within an unstructured mesh as shown in Figure 30(b). However, errors are unavoidable when numerical integrations are enforced over Ω_h instead of Ω . As a result the boundary conditions are applied on the outline of $\partial\Omega_h$, which can induce a significant loss of accuracy as will be shown in Chapter 4. Once we are capable of covering the domain with curved elements of high order, we obtain a discrete domain $\Omega_{h,p}$ and the representation is largely improved as illustrated in Figure 30(c).

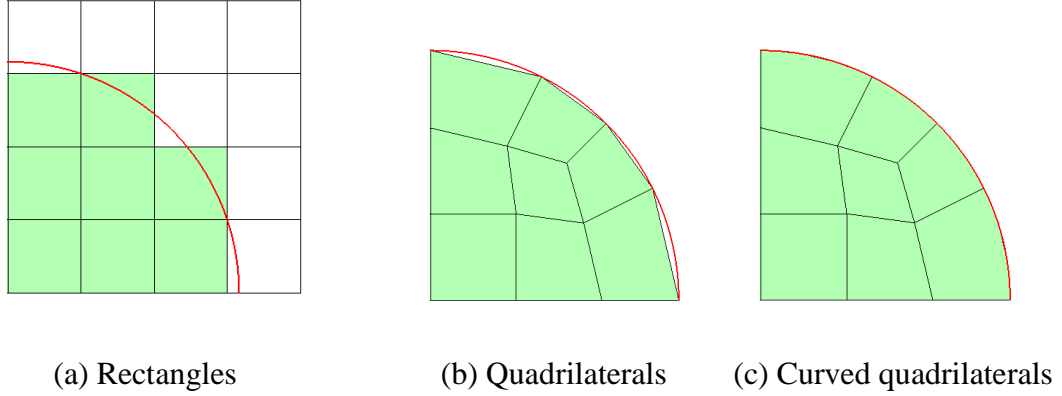


Figure 30: Discretization of a domain consisting of a quarter circle (closed by red arc and two boundaries), the discrete domains Ω_h with (a) rectangles, (b) quadrilaterals and (c) the approximation domain $\Omega_{h,p}$ with curved-edge quadrilaterals.

Hence, we apply the curved elements introduced in the preceding chapter and cover the computational domain with N non-overlapping elements. For a given element K , there exists a bijective mapping from reference element \hat{K} , as introduced above, to element K . The finite element space is a linear combination of the basis functions up to the order of p . A suitable function space of polynomials $V_{N_p} = \text{span}(\varphi_1, \varphi_2, \dots, \varphi_{N_p})$ is defined on the reference element \hat{K} . The time and space continuous electromagnetic field quantities are approximated locally in each element:

$$\mathbf{U}_h(\mathbf{r}, t) = \sum_{i=1}^{N_p} \mathbf{u}_i(t) \varphi_i(\mathbf{r}), \quad (3-33)$$

with Legendre polynomial basis functions φ_i of order $i \in \{1, 2, \dots, N_p\}$. The time-dependent coefficients $\mathbf{u}_i(t)$ are the numerical Degrees of Freedom (DoF). As the basis functions have element-wise compact support the approximation is to be understood in the element-wise sense as well.

It is assumed that materials are isotropic and have constant parameters within each element $\alpha = \begin{bmatrix} \varepsilon & 0 \\ 0 & \mu \end{bmatrix}$. Inserting the generic approximation (3-33) into the conservative form (3-29), multiplying by a test function φ_j of order $j \in \{1, 2, \dots, N_p\}$ and integrating over element K yields:

$$\int_K \alpha \frac{\partial \mathbf{U}_h}{\partial t} \varphi_j d\mathbf{r} + \int_K \nabla \cdot \mathbf{F}(\mathbf{U}_h) \varphi_j d\mathbf{r} = 0. \quad (3-34)$$

Since Galerkin testing is applied, the test functions are identical to the basis functions. We introduce the numerical flux $\mathbf{F}^*(\mathbf{U}_h)$, which depends on the two local solutions of the

neighboring elements at the interface. Integrating equation (3-36) by parts one obtains:

$$\int_K \alpha \frac{\partial \mathbf{U}_h}{\partial t} \varphi_j d\mathbf{r} - \int_K \mathbf{F}(\mathbf{U}_h) \nabla \varphi_j d\mathbf{r} + \int_{\partial K} \mathbf{n} \cdot \mathbf{F}^*(\mathbf{U}_h) \varphi_j d\mathbf{r} = 0, \quad (3-35)$$

where \mathbf{n} denotes the outward pointing normal vector on the boundary of element K . The numerical flux has to be introduced as the approximation is double-valued at every point on element interfaces. This is a consequence of the compact support and has to be resolved before the surface integration in equation (3-35) can be evaluated.

For a certain pair of basis and test polynomial orders (i, j) , where $1 \leq i, j \leq N_p$, equation (3-35) in element K reads:

$$M_K^{ij} \frac{\partial \mathbf{u}_i}{\partial t} + S_K^{ij} \mathbf{F}(\mathbf{u}_i) + R_K^{ij} \mathbf{F}(\mathbf{u}_i) = 0, \quad (3-36)$$

$$M_K^{ij} = \alpha \int_K \varphi_i(\mathbf{r}) \varphi_j(\mathbf{r}) d\mathbf{r}, \quad (3-37)$$

$$S_K^{ij} = - \int_K \varphi_i(\mathbf{r}) \nabla \varphi_j(\mathbf{r}) d\mathbf{r}, \quad (3-38)$$

$$R_K^{(i,i')j} = \int_{\partial K} \mathbf{n} \varphi_{i,i'}^*(\mathbf{r}) \varphi_j(\mathbf{r}) d\mathbf{r}. \quad (3-39)$$

The above equations introduce the mass term M_K^{ij} , the stiffness term S_K^{ij} and the flux term $R_K^{(i,i')j}$, which is also called Riemann solver. The unified value at the interface of element i and its neighbor i' is denoted by $\varphi_{i,i'}^*$. The volume integral terms in equations (3-37), (3-38) and the surface integration term in equation (3-39) can be computed with, for example, Legendre–Gauss–Lobatto quadrature rules. Instead of evaluating the above integrals on the possibly curved elements in the physical domain, the mapping function introduced in Chapter 2 is applied for transforming every element back to the standard hexahedral element \widehat{K} defined in the reference domain $\xi = \{(\xi, \eta, \gamma) | (\xi, \eta, \gamma) \in [-1, 1]^3\}$.

$$M_K^{ij} = \alpha \int_{\widehat{K}} \varphi_i(\xi) \varphi_j(\xi) |J(\xi)| d\xi, \quad (3-40)$$

$$S_K^{ij} = - \int_{\widehat{K}} \varphi_i(\xi) \nabla \varphi_j(\xi) |J(\xi)| d\xi, \quad (3-41)$$

$$R_K^{(i,i')j} = \int_{\partial\widehat{K}} \mathbf{n} \varphi_{i,i'}^*(\boldsymbol{\xi}) \varphi_j(\boldsymbol{\xi}) |J(\boldsymbol{\xi})| d\xi. \quad (3-42)$$

As discussed in Section 2.3.1 in the last chapter, $|J(\boldsymbol{\xi})|$ is the Jacobian of an invertible transformation from the reference domain to a given hexahedral element in the computational domain. The basis functions are defined as a tensor product of Legendre polynomials on the reference element and take a different form in the physical elements, which is given by the mapping. The basis functions can, hence, take a different form on each physical element. Among the different choices for the flux [75] we employ the central flux for $R_K^{(i,i')j}$ yielding:

$$R_K^{(i,i')j} = R_{K\xi}^{(i,i')j} + R_{K\eta}^{(i,i')j} + R_{K\gamma}^{(i,i')j}, \quad (3-43)$$

$$R_{K\xi}^{(i,i')j} = \int_{\partial\widehat{R\xi}} n_\xi \cdot \left[\frac{\varphi_i(\pm 1, \eta, \gamma) + \varphi_{i'}(\mp 1, \eta, \gamma)}{2} \right] \cdot \varphi_j(\pm 1, \eta, \gamma) |J(\pm 1, \eta, \gamma)| d\eta d\gamma, \quad (3-44)$$

$$R_{K\eta}^{(i,i')j} = \int_{\partial\widehat{K\eta}} n_\eta \cdot \left[\frac{\varphi_i(\xi, \pm 1, \gamma) + \varphi_{i'}(\xi, \mp 1, \gamma)}{2} \right] \varphi_j(\xi, \pm 1, \gamma) |J(\xi, \pm 1, \gamma)| d\xi d\gamma, \quad (3-45)$$

$$R_{K\gamma}^{(i,i')j} = \int_{\partial\widehat{K\gamma}} n_\gamma \cdot \left[\frac{\varphi_i(\xi, \eta, \pm 1) + \varphi_{i'}(\xi, \eta, \mp 1)}{2} \right] \varphi_j(\xi, \eta, \pm 1) |J(\xi, \eta, \pm 1)| d\xi d\eta. \quad (3-46)$$

Unlike other fluxes such as full or partial upwinding fluxes, the central flux leads to an energy conserving scheme [74], which can be efficiently integrated in time using the Leapfrog time integrator. This time integrator is the subject of the following section.

3.4.1. System matrix in conservative form

The element-wise mass term M_K^{ij} , stiffness term S_K^{ij} and flux term $R_K^{(i,i')j}$ defined in the last section are ready to be assembled into global matrices. As a reminder, a pair of indices (i, j) is referred to the orders of basis and test functions. The index of element K is represented by a number k , which is in the region $1 \leq k \leq N$. We assemble the mass,

stiffness and flux terms into the matrices:

$$[\mathbf{M}]_{i,j}^k = M_K^{ij}, \quad (3-47)$$

$$[\mathbf{P}]_{(i,i'),j}^k = S_K^{ij} + R_K^{(i,i')j}, \quad (3-48)$$

where k on the left hand side indicates the number of the block. The mass matrix \mathbf{M} is a block diagonal matrix. The derivative matrix \mathbf{P} has a block pattern as well, however, it is not block diagonal due to the contributions of the element neighbors i' . One block has N_p rows and N_p columns, where p is the maximum order. In three dimensions for example, N_p equals to $(p+1)^3$ for tensor product hexahedral element.

In the beginning of section 3.4.1., we have defined the conservative form of Maxwell's equations (3-29) (3-30) (3-31). The electric and magnetic fields are denoted as \mathbf{U} . As a result, the DoF in equation (3-33) can be assembled into the vector:

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N_p}). \quad (3-49)$$

The matrix form with the symbols $[\mathbf{M}]_{i,j}^k$ and $[\mathbf{P}]_{(i,i'),j}^k$ can be written as follows:

$$\mathbf{M} \frac{d}{dt} \mathbf{u} + \mathbf{P} \cdot \mathbf{F}(\mathbf{u}) = 0. \quad (3-50)$$

Although the flux is written into x - y - and z -components as equation (3-43) shows, the matrix form (3-50) is in general not limited to certain coordinates. For simplicity, we focus on Cartesian coordinates for the current approach. Therefore the operators can be written explicitly as:

$$\mathbf{P} = (\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z), \quad (3-51)$$

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_\varepsilon & 0 \\ 0 & \mathbf{M}_\mu \end{pmatrix}, \quad (3-52)$$

$$\mathbf{M}_\varepsilon = \begin{pmatrix} \mathbf{M}_\varepsilon^x & 0 & 0 \\ 0 & \mathbf{M}_\varepsilon^y & 0 \\ 0 & 0 & \mathbf{M}_\varepsilon^z \end{pmatrix}, \quad (3-53)$$

$$\mathbf{M}_\mu = \begin{pmatrix} \mathbf{M}_\mu^x & 0 & 0 \\ 0 & \mathbf{M}_\mu^y & 0 \\ 0 & 0 & \mathbf{M}_\mu^z \end{pmatrix}. \quad (3-54)$$

In equation (3-30), we replace the divergence operator \mathbf{P} on the left hand side and curl operator \mathbf{C} on the right hand side, relations become:

$$\mathbf{P} \cdot \mathbf{F}(\mathbf{U}) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{C} \cdot \mathbf{E} \\ \mathbf{C} \cdot \mathbf{H} \end{pmatrix}, \quad (3-55)$$

where the curl operator \mathbf{C} in Cartesian coordinates is:

$$\mathbf{C} = \begin{pmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{pmatrix}. \quad (3-56)$$

We separate the electric and magnetic field vectors \mathbf{e} and \mathbf{h} from \mathbf{u} , and define the coefficients the similar way as equation (3-49), the differential matrix form of Maxwell's equations [74] can be obtained as follow:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{M}_\varepsilon \mathbf{e} \\ \mathbf{M}_\mu \mathbf{h} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & -\mathbf{C} \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ \mathbf{h} \end{pmatrix} = \mathbf{0}. \quad (3-57)$$

The two systems of matrix (3-50) and (3-57) are equivalent to each other, only expressed in different mathematic formulations. We discretize the system of equations (3-57) in time. A frequently used time integration scheme is the so-called Leapfrog scheme, which is based on second order central differences. The electric and magnetic field values are allocated at different time steps, shifted half a time step from each other:

$$\mathbf{e}_{n+1} = \mathbf{e}_n + \Delta t (\mathbf{M}_\varepsilon^{-1} \mathbf{C}) \mathbf{h}_{n+1/2}, \quad (3-58)$$

$$\mathbf{h}_{n+3/2} = \mathbf{h}_{n+1/2} - \Delta t (\mathbf{M}_\mu^{-1} \mathbf{C}) \mathbf{e}_{n+1}. \quad (3-59)$$

The choice of a smaller time step Δt leads to more accurate results. The scheme is conditionally stable and non-dissipative. The energy conservation is the method is shown in [74, 76]. The stability of the leapfrog method is ensured by the Courant-Friedrich-Lewy (CFL) criterion [77]. In the case of DG with piecewise constant basis functions ($p = 0$) we assume the space increment is $(\Delta x_k, \Delta y_k, \Delta z_k)$ in each coordinate and the time step Δt , where k is the local grid cell. Then the time step Δt has to satisfy the stability condition:

$$\Delta t \leq \Delta t_{max} \leq \min \left\{ \frac{1}{c_k} \cdot \frac{1}{\sqrt{\left(\frac{1}{\Delta x_k}\right)^2 + \left(\frac{1}{\Delta y_k}\right)^2 + \left(\frac{1}{\Delta z_k}\right)^2}} \right\}. \quad (3-60)$$

The speed of light c_k is determined by the material properties of the local element. With the electric permittivity ε_k and magnetic permeability μ_k , we have:

$$c_k = \frac{1}{\sqrt{\varepsilon_k \mu_k}}. \quad (3-61)$$

The grid dispersion equation provides a way to derive the stability criterion:

$$\left(\frac{\sin(k_x \Delta x)}{\Delta x}\right)^2 + \left(\frac{\sin(k_y \Delta y)}{\Delta y}\right)^2 + \left(\frac{\sin(k_z \Delta z)}{\Delta z}\right)^2 = \frac{1}{c^2} \left(\frac{\sin(\omega \frac{\Delta t}{2})}{\frac{\Delta t}{2}}\right)^2. \quad (3-62)$$

In the DGM background, the interpolation order needs to be taken into account. With the typical length of $(\Delta x_k, \Delta y_k, \Delta z_k)$ being reduced, the time step Δt should be even smaller. Since the Leapfrog method is only second order accurate, the high order accuracy in space can be dominated by the time discretization error. In order to show high order convergence in space, we choose very fine time steps. Only Leapfrog time integration is considered in this thesis. Nevertheless, it is worth to mention that the time integration can also be of high order. Some available high order schemes for time integration can be found in [75] for the Runge Kutta method and in [78] for high order symplectic integration.

4. Eigenvalue Problems in Electromagnetics

In this chapter the mesh scheme proposed in Chapter 2 is applied with the Discontinuous Galerkin Method described in the preceding chapter for the simulations of electromagnetic problems. The geometries in the computational domain include curved boundaries, and it is shown that the full accuracy potential of the high order method can be exploited if body-fitting curved meshes are applied.

First a number of two-dimensional examples are considered for showing the general properties of the method. These properties include the convergence order in the presence of curved objects and boundaries but also spectral properties.

The analysis of two-dimensional problems is still an important issue, in particular because very accurate 2D solutions are required in the analysis of three-dimensional problems. In particular, electromagnetic problems of long homogeneous systems can be seen as a two-dimensional eigenvalue problem [80]. The more important application, however, are waveguide ports, which are a concept of general importance in numerical electromagnetic analysis [81]. They are used for guiding excitation fields into the domain of interest and for extracting the corresponding fields at the structure outlets. Waveguide ports are usually placed at one or more locations at the domain boundary. The accuracy of the simulation of the three-dimensional problem critically depends on accurate solutions of the excitation modes. Body-fitting curved mesh can be applied to improve the accuracy of the port modes, where the resonant frequencies are exactly the eigenvalues of the two-dimensional area enclosed by the port.

When the port area is properly defined to cover the cross sections of a waveguide, the propagation modes can be calculated. In case the cut-off frequency is exceeded, their respective electric and magnetic modes will be excited and propagate along the longitudinal direction. Therefore, the essential part is to solve the waveguide ports as a two dimensional eigenvalue problem. Following the port mode calculation the three dimensional problem can be considered.

In this chapter, validations of the DG method are first made with a square cavity problem, where no errors arise from approximating the domain but from the numerical field solution only. In order to investigate the geometric errors, we choose a disc shaped domain. There, different strategies to construct the mesh are shown to make a big difference in terms of overall numerical errors.

Furthermore, numerical results are compared to analytical solutions given by solving Helmholtz's equations in cylindrical coordinates for a circular metallic cavity. We compare one of the transverse modes and measure the L_2 -errors by means of the distance in L_2 -norm between the numerical results and their respective analytical ones.

For evaluation of the convergence properties of the DG method with body-fitting curved meshes, L_2 -errors are observed while increasing the number of mesh elements. Optimal convergence behavior is observed, where the slope of the error reduction is one order higher than the polynomial degree. In other words, the speed of approaching the exact solution is accelerated by increasing the polynomial order ideally.

In the end, transient phenomena of electromagnetic waves are simulated. The purpose is to solve eigenvalue problems using a time-domain approach [82] by recording the field strength signal with an electric probe at each time step. Field variations represent the process of wave propagation in the domain including reflections at the boundary. As a result, the Fourier transformation of the time signal provides a spectrum, which shows the resonant frequencies of the given geometry.

4.1. High Order Accuracy in a Square Domain

As a first test to examine the accuracy of the method, we consider a two dimensional cavity problem in a square domain $[-1, 1]^2$ with PEC boundaries. The example is a verification of the DG implementation proposed in Chapter 3.

The exact reference solution in the square cavity is obtained by the Maxwell's equations for the two-dimensional case. Let us consider the Transverse Electric (TE) case with fields component (E_x, E_y, H_z) in Cartesian coordinates. The symbols are defined previously in Chapter 3, yielding:

$$\varepsilon \frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y}, \quad (4-1)$$

$$\varepsilon \frac{\partial E_y}{\partial t} = -\frac{\partial H_z}{\partial x}, \quad (4-2)$$

$$\mu \frac{\partial H_z}{\partial t} = -\frac{\partial E_y}{\partial x} + \frac{\partial E_x}{\partial y}. \quad (4-3)$$

With PEC boundary conditions, the TE solutions are:

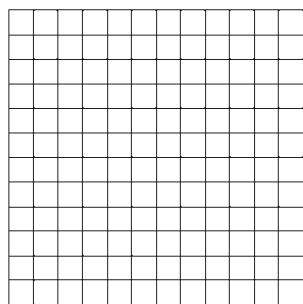
$$E_x(x, y, t) = \frac{\pi n}{\omega} \cos(m\pi x) \sin(n\pi y) \sin(\omega t), \quad (4-4)$$

$$E_y(x, y, t) = -\frac{\pi n}{\omega} \sin(m\pi x) \cos(n\pi y) \sin(\omega t), \quad (4-5)$$

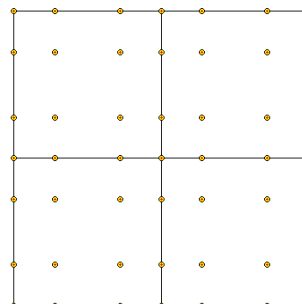
$$H_z(x, y, t) = \cos(m\pi x) \cos(n\pi y) \cos(\omega t). \quad (4-6)$$

In Figure 31(a) and (b), two types of elements are constructed for the schemes with piecewise constant and piecewise cubic approximations, respectively. The first mesh (a) represents field distributions as a combination of constant values within 144 elements, while the second mesh (b) interpolates field values using a patch of interpolation points within each of the 4 elements. There are 144 DoF in (a), but only 64 DoF are employed in (b). The corresponding initial field distributions are presented in Figure 31(c) and (d), where the mesh using piecewise cubic elements yields a much smoother approximation than that with piecewise constant elements. Figure 31(e) and (f) shows the distributions of point-wise evaluated errors at the end of one period of oscillation. Again, the high order mesh leads to better performance, where the peak value of the errors is about 8×10^{-3} . In comparison the maximum error computed by the piecewise constant mesh is above 2.5×10^{-1} .

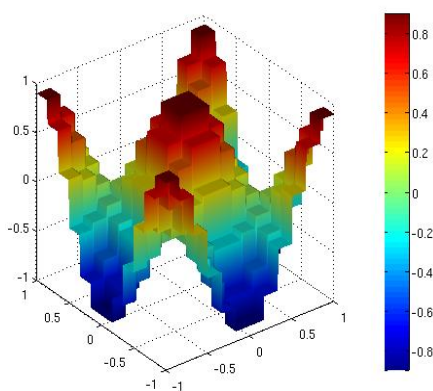
In order to measure the influence of the basis order, we show L_2 -errors calculated with orders of one to four. In Figure 32, the error values are depicted versus the square root of DoF number. The slope of the error lines indicates the rates of convergence in the L_2 -norm. The convergence rate is increased ideally along with the polynomial order.



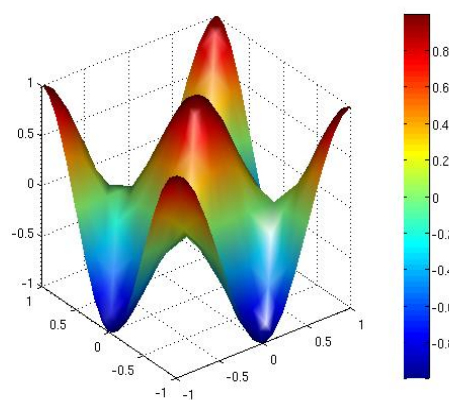
(a) h-refinement with 144 elements approximated by piecewise constant polynomials



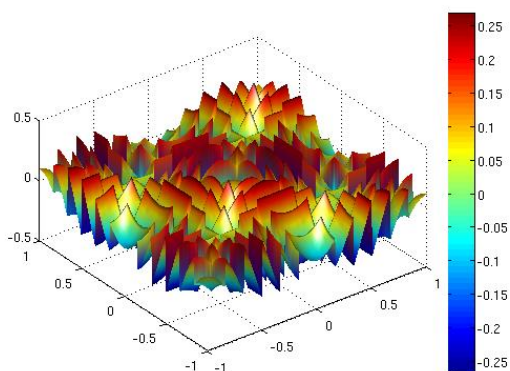
(b) p-refinement with 4 elements approximated by piecewise cubic polynomials



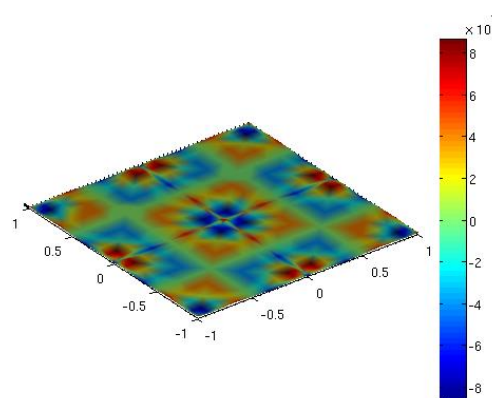
(c) Initial field distribution with piecewise constant polynomials



(d) Initial field distribution with piecewise cubic polynomials



(e) Error distribution with piecewise constant polynomials



(f) Error distribution with piecewise cubic polynomials

Figure 31: A square domain is covered with (a) h-refined and (b) p-refined meshes, on which (c) (d) the initial electric field distributions are based. After discretization with the DG method errors as plotted in (e) (f) are obtained for 0th and 3rd order polynomials.

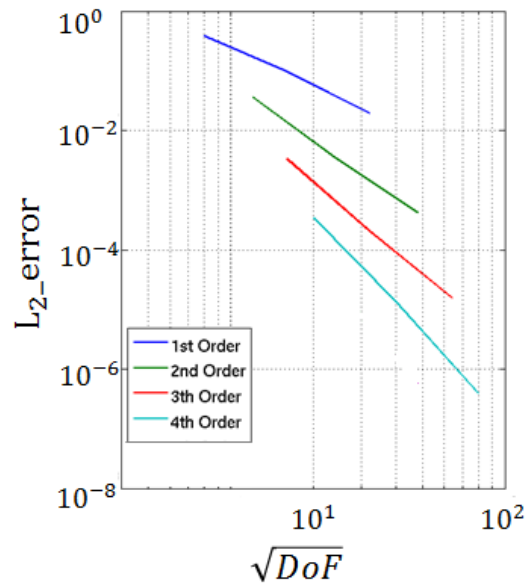


Figure 32: L_2 error of electric field with polynomial increasing from 1st to 5th order. The errors show $(p+1)$ convergence slopes in a logarithmic plot.

4.2. Electromagnetic Fields in a Cylindrical Domain

In order to take geometric errors into consideration, we further solve the problem for geometries with curved features. The central question is whether the high order accuracy in the square geometry can be maintained in the curved domain. The numerical tests will be based on the high order mesh scheme that has been proposed in Chapter 2.

4.2.1 Electromagnetic Solutions in Cylindrical Coordinates

In a cylindrical metallic waveguide, the propagating modes can be solved in a circular cross section. In order to evaluate errors of the numerical computation in time domain, the problem is first calculated analytically [80]. Consider that the domain of interest is source-free. The solution is obtained by solving the Helmholtz equations in frequency domain:

$$\nabla^2 \mathbf{E} + k^2 \mathbf{E} = 0, \quad (4-7)$$

$$\nabla^2 \mathbf{H} + k^2 \mathbf{H} = 0, \quad (4-8)$$

where $k^2 = \omega^2 \mu \epsilon - j \omega \mu \sigma$, with the permittivity ϵ , permeability μ and conductivity σ . In the case of low-loss media and high frequency, $\sigma \ll \omega \epsilon$, i.e., $\sigma \approx 0$,

$$k = \omega\sqrt{\mu\varepsilon}. \quad (4-9)$$

Assuming a linear, isotropic and non-dispersive material in the domain, the electric and magnetic components in cylindrical coordinates are denoted as (E_ρ, E_ϕ, E_z) and (H_ρ, H_ϕ, H_z) respectively. Solutions of the Helmholtz equations are given by:

$$E_\rho = -j\beta \frac{\partial U}{\partial \rho} - \frac{j\omega\mu}{\rho} \frac{\partial V}{\partial \phi}, \quad (4-10)$$

$$E_\phi = -\frac{j\beta}{\rho} \frac{\partial U}{\partial \phi} + j\omega\mu \frac{\partial V}{\partial \rho}, \quad (4-11)$$

$$E_z = (k^2 - \beta^2)U = T^2U, \quad (4-12)$$

$$H_\rho = -j\beta \frac{\partial V}{\partial \rho} + \frac{j\omega\varepsilon}{\rho} \frac{\partial U}{\partial \phi}, \quad (4-13)$$

$$H_\phi = -\frac{j\beta}{\rho} \frac{\partial V}{\partial \phi} - j\omega\varepsilon \frac{\partial U}{\partial \rho}, \quad (4-14)$$

$$H_z = (k^2 - \beta^2)V = T^2V, \quad (4-15)$$

where U and V are the so-called Borgnis' functions [80]. They are often used for reducing vector Helmholtz equations into scalar Helmholtz equations in cylindrical and spherical coordinate systems. The longitudinal phase coefficient is $\beta = k_z = \sqrt{k^2 - T^2}$, where k is defined in equation (4-9) and T is the transverse angular wave number given by:

$$T = \sqrt{k_x^2 + k_y^2} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}, \quad (4-16)$$

(1) For Transverse Magnetic (TM) modes: $V = 0$, $U \neq 0$ this yields,

$$U(\rho, \phi, z) = U_0 J_n(T\rho) \cos(n\phi) e^{-j\beta z}. \quad (4-17)$$

Applying equation (4-17) with the boundary condition $U|_{\rho=a} = 0$, we have $J_n(Ta) = 0$. Therefore, the wave number and frequency for TM modes are:

$$T_{\text{TM}_{nm}} = \frac{x_{nm}}{a}, \quad (4-18)$$

$$f_{\text{cTM}_{nm}} = \frac{c}{2\pi\sqrt{\mu_r\varepsilon_r}} \frac{x_{nm}}{a}, \quad (4-19)$$

where x_{nm} is the m -th root of the Bessel function of n -th order $J_n(x) = 0$. With c the speed of light, μ_r relative permeability and ε_r relative permittivity, $f_{\text{cTM}_{nm}}$ is obtained as the cutoff frequency of TM modes. Combining equation (4-17) with equation (4-12), the z -components of the electric field can be written as:

$$E_z = T^2 U_0 J_n(T\rho) \cos(n\phi) e^{-j\beta z}. \quad (4-20)$$

(2) Similarly, for Transverse Electric (TE) modes, $U = 0$, $V \neq 0$:

$$V(\rho, \phi, z) = V_0 J_n(T\rho) \cos(n\phi) e^{-j\beta z}. \quad (4-21)$$

Applying equation (4-21) with the boundary condition $\frac{\partial V}{\partial \rho} \big|_{\rho=a} = 0$, we have $J'_n(Ta) = 0$. Therefore, the wave number and frequency for TE modes are:

$$T_{\text{TE}_{nm}} = \frac{y_{nm}}{a}, \quad (4-22)$$

$$f_{\text{cTE}_{nm}} = \frac{c}{2\pi\sqrt{\mu_r\epsilon_r}} \frac{y_{nm}}{a}, \quad (4-23)$$

where y_{nm} is the m -th root of the derivative of the Bessel function of n -th order $J'_n(y) = 0$. Combining equation (4-21) with equation (4-15), the z -component of magnetic fields can be written as:

$$H_z = T^2 V_0 J_n(T\rho) \cos(n\phi) e^{-j\beta z}, \quad (4-24)$$

For convenience the roots of the Bessel function x_{nm} and the roots of the derivative of the Bessel function are given in Appendix A.

4.2.2 Wave Propagation in a Circular Domain

This second example deals with wave propagation within a circular domain with PEC boundary conditions. Its purpose is to give an immediate impression and illustration of the importance of curved elements for the treatment of curved boundaries.

As illustrated in Figure 33(a)-(c), the circular domain is discretized with three meshes. The meshes consist of 12 and 21 straight-sided as well as 21 curved elements, respectively. The solution approximation order of all elements is six. With h -refinement of the mesh, the accuracy can be increased linearly with the number of elements only. This is suboptimal as raising the number of elements is computationally expensive for high order computations. From Figure 33 (b) to (c), a r -refinement defined in Section 1.1.2 is applied such that the geometrical approximation order matches the element order.

In Figure 33(d)-(f) and (g)-(i) snapshots at two instances in time are depicted, where the excitation is a sharp pulse of the electric field in the center of the domain. Wave propagation occurs cylindrically. Due to the symmetric shape of the exact circular domain, the wave should still be cylindrical after reflection from the boundary as shown in (i). However, this is not the case for (g) and (h), where the meshes are composed with straight-sided elements. In this case, the wave reflections at the corners of the piecewise linear boundaries give rise to oscillations, which are unphysical in a circular domain with smooth curved boundary.

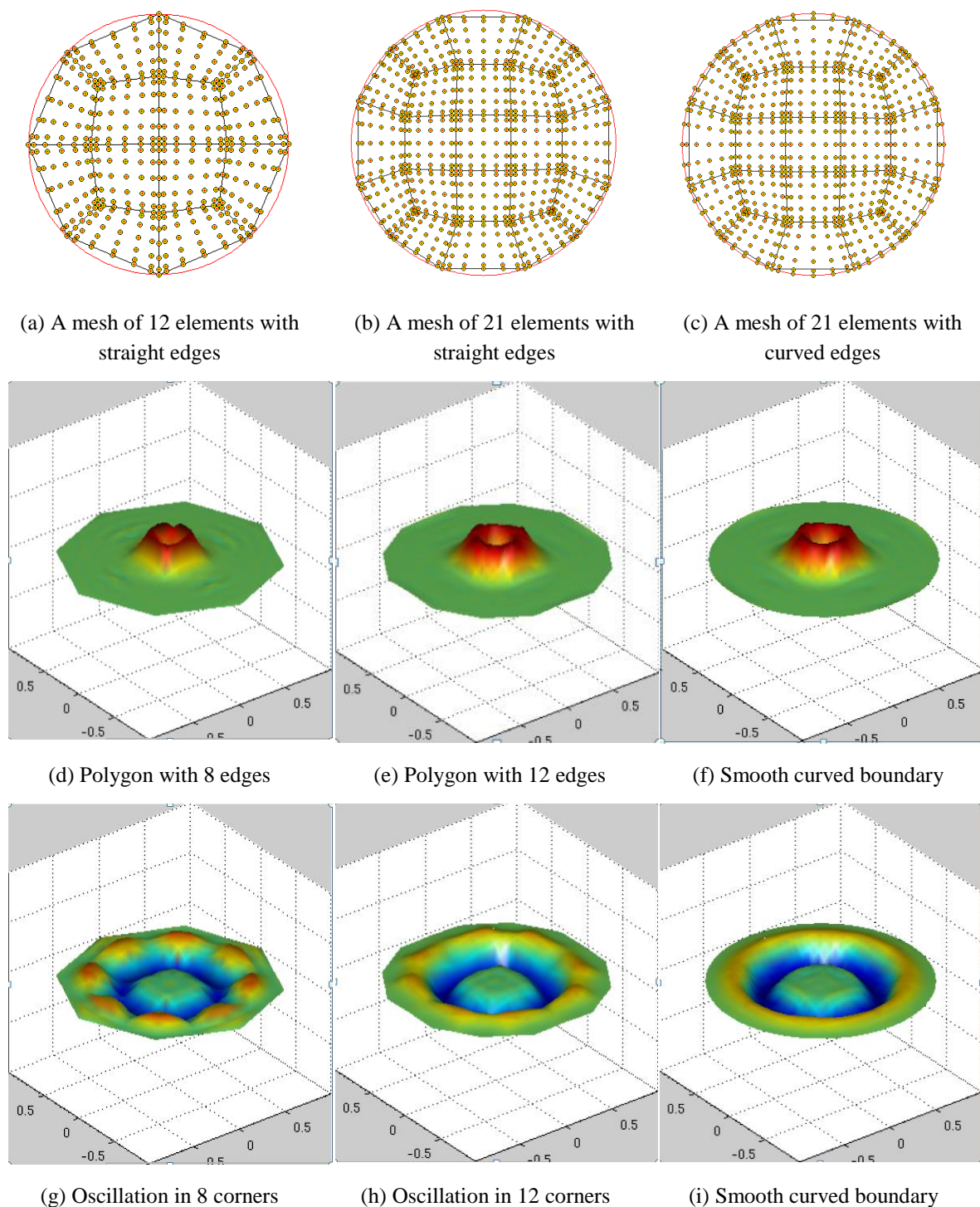


Figure 33: A circular domain is discretized with three meshes with interpolation points of 6th order (a)-(c). The computation is performed by the central flux DG method with 6th order polynomial basis. For the polygonal and smooth domains excitation occurs in the center. The resulting electric fields before and after reflection are depicted in (d)-(f) and (g)-(i). Only for curved elements, the wave remains cylindrical after reflection.

4.2.3 High Order Convergence

In the preceding section a more qualitative image of the advantages of curved meshes was given. Now, a quantitative study is performed showing how the approximation error of the geometry is linked to the convergence rate of the complete scheme.

Consider transverse magnetic (TM) problems in a two-dimensional circular domain. In this DG approach, the explicit leap-frog scheme is used for the time discretization [43]. For a convergence study, the TM_{31} mode in a circular cavity is chosen. The errors are measured in the L_2 -norm after one periodic oscillation. The setup is considered also in [6], where triangular elements are employed. Nevertheless, this allows for a comparison.

In the first step, we solve an resonant problem of TM_{31} mode. The z -component of the electric field distribution is set as the initial conditions. With time evolution, the electric field of TM_{31} mode shows periodic oscillations in the cavity. The first half period is illustrated in Figure 34(a)-(f) with six snapshots.

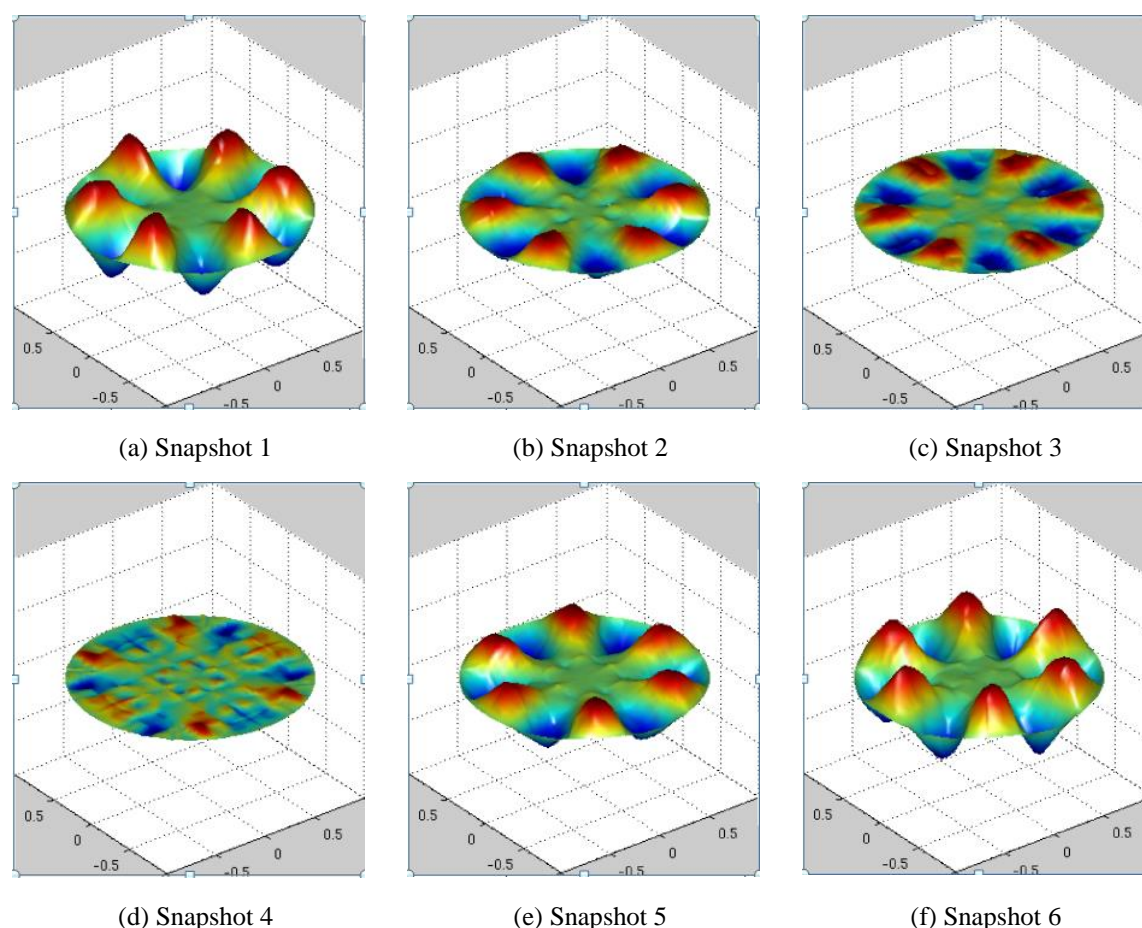


Figure 34: Snapshots of the electric field distribution of the TM_{31} mode in half a period of oscillation. The computation is performed by the central flux DG method with 56 elements of 6th order polynomial basis.

The circular domain in has Figure 34 been decomposed with 56 elements including curved elements. Based on the discretised domain, the system of equations (4-1)-(4-3) has been solved with a central flux DG scheme using a 6th order polynomial basis.

In the next step, we evaluate the convergence properties of the overall method with body-fitting meshes. Convergence rates are obtained by computing L_2 -errors on a series of meshes while increasing the number of mesh elements. The optimal convergence rate is one order higher than the polynomial degree. In this case, convergence rate increases along with each increase of the polynomial order.

In the following convergence rates are obtained numerically using a set of three meshes. All boundary elements are fitted to the boundary with the TFI technique. The meshes are depicted in Figure 35 (a) for 3rd order elements.

The analytical solution of the electric field has been computed according to equation (4-20) (4-21) and plotted in Figure 35(b).

The rates of error convergence are measured with basis functions from 1st to 6th degree. In the double logarithmic plot, the slope of the error reduction curve (i.e. the convergence rate) depends on the ratio of errors on consecutively refined meshes the corresponding numbers of DoF. The dashed triangles in Figure 35(c) show that the 1st, 2nd and 6th order bases achieve 2nd, 3rd and 7th convergence order. The figure does therefore show that optimal convergence orders of $(p+1)$ are obtained.

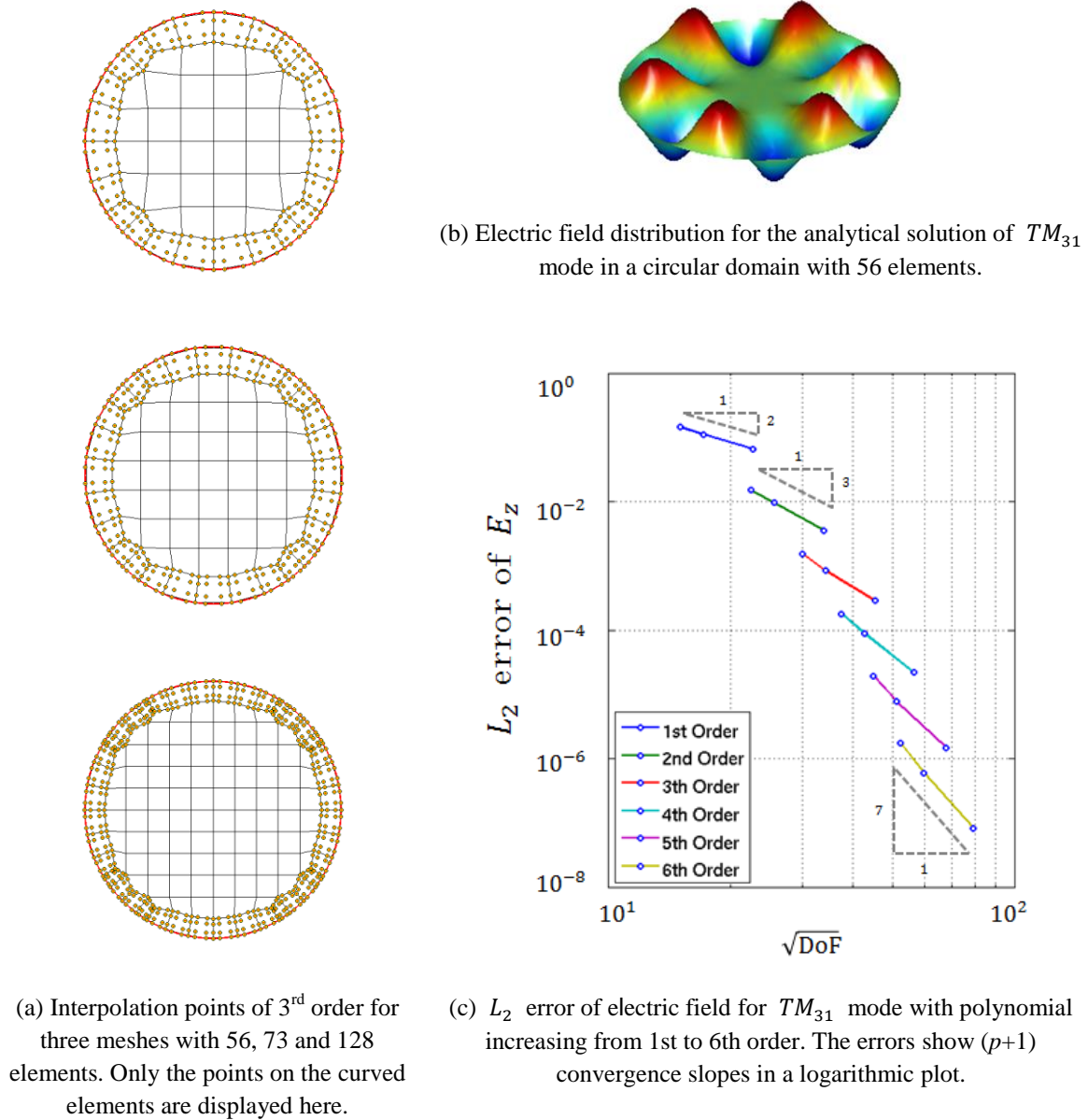


Figure 35: A resonant TM_{31} mode in the circular cavity is calculated on (a) three sets of body-fitting meshes. The z-component of the electric field (b) is plotted with the analytical solutions. The numerical solutions obtained by the DG method with upwind flux (c) shows $(p+1)$ convergence orders.

The history data of errors and convergence orders is shown in Table 4-1, Table 4-2 and Table 4-3, where the latter two cases yield optimal convergence. In the first table, only straight sided elements are used, as a result the final errors are limited by linear representation of the boundary. The convergence rate is dominated by the geometric errors, and shows no improvement by increasing the order of the basis. In the second table, we fix the quadrature order as 10, so that the geometric errors have the equal

amount for different polynomial p . This second choice shows optimal convergence of order $(p+1)$. So a practical question can be which quadrature order is accurate enough for a given polynomial order p . In the last case Table 4-3, we choose the same order for both quadrature and solution basis, which is a common routine in the spectral element method [79]. Except for the first order basis, $(p+1)$ convergence is achieved for all higher order basis.

It can be summarized that high order numerical methods achieve their full accuracy only if the geometric approximation matches the field approximation order. Figure 35 shows that while the mesh conforms to the circular boundary, its interior part remains fully structured. This is an important advantage, as the inclusion of deformation information into the mass and stiffness matrices via Jacobian as shown in equations (3-40)-(3-41) make their evaluation computationally much more expensive. Additionally, the simplified evaluation of fluxes across interfaces aligned with Cartesian coordinates is preserved in the interior.

Table 4-1 Convergence rates and L_2 -errors measured after one period of oscillation. Only elements with straight edges have been used. The quadrature order is fixed as 10.

Polynomial order (p)	Mesh_1	Mesh_2	Mesh_3	Slope (rate)
1	1.48E-01	1.14E-01	6.63E-02	1.94
2	1.87E-02	1.27E-02	5.47E-03	2.98
3	1.21E-02	9.46E-03	4.46E-03	2.45
4	1.24E-02	9.52E-03	4.49E-03	2.50
5	1.25E-02	9.49E-03	4.52E-03	2.48
6	1.25E-02	9.51E-03	4.53E-03	2.48

Table 4-2 Convergence rates and L_2 -errors measured after one period oscillation. There are curved elements on the boundaries. The quadrature order is fixed as 10.

Polynomial order (p)	Mesh_1	Mesh_2	Mesh_3	Slope (rate)
1	1.45E-01	1.11E-01	6.56E-02	1.90
2	1.50E-02	9.56E-03	3.57E-03	3.48
3	1.53E-03	8.39E-04	2.85E-04	4.03
4	1.78E-04	8.80E-05	2.18E-05	5.06
5	1.92E-05	7.71E-06	1.46E-06	6.18
6	1.73E-06	6.02E-07	8.19E-08	7.33

Table 4-3 Convergence rates and L_2 -errors measured after one period oscillation. The curved elements on the boundaries and the quadrature order is p.

Polynomial order (p)	Mesh_1	Mesh_2	Mesh_3	Slope (rate)
1	2.75E-01	2.58E-01	1.98E-01	0.81
2	4.36E-02	3.11E-02	1.20E-02	3.17
3	4.61E-03	2.38E-03	8.79E-04	3.93
4	4.60E-04	2.36E-04	6.01E-05	4.92
5	4.85E-05	2.01E-05	3.91E-06	6.05
6	4.33E-06	1.54E-06	2.09E-07	7.21

4.3. Frequency Spectrum Analysis

In Section 4.2.3, the simulations considered one mode of the spectrum only. However, time-domain methods are usually employed for computing the solution in wide band of frequencies. Nevertheless, the extraction of a large number of eigenmodes is still a challenging problem. One solution is found in performing a broad-band time-domain simulation and extracting the resonant frequencies using a subsequent Fourier transformation.

In order to obtain precise resonant frequencies two important conditions have to be met. First, the time-domain simulation has to be sufficiently long for generating a number of samples that will allow for the desired resolution of the frequency spectrum, and second the geometry of the problem at hand has to be represented properly. Eigenvalue calculations are known to be particularly sensitive to geometric modeling errors as they immediately translate into resonance shifts. Hence, the main steps consist of geometry modeling, transient wave solving and Fourier analysis.

The geometry is a unit circular domain with PEC boundary conditions. The excitation is a broad-band pulse with a Gaussian profile in time. It is placed in an off-center position in order to avoid zeros of low order resonant modes. In Figure 36(a)-(h) a number of snapshots is depicted showing the excellent reflection behavior from the curved boundary.

The mesh on the boundary has been equipped with curved elements for the simulation in Figure 36. With a total of 32 elements of fifth order, the transient wave propagation has been solved by the DG method with central flux.

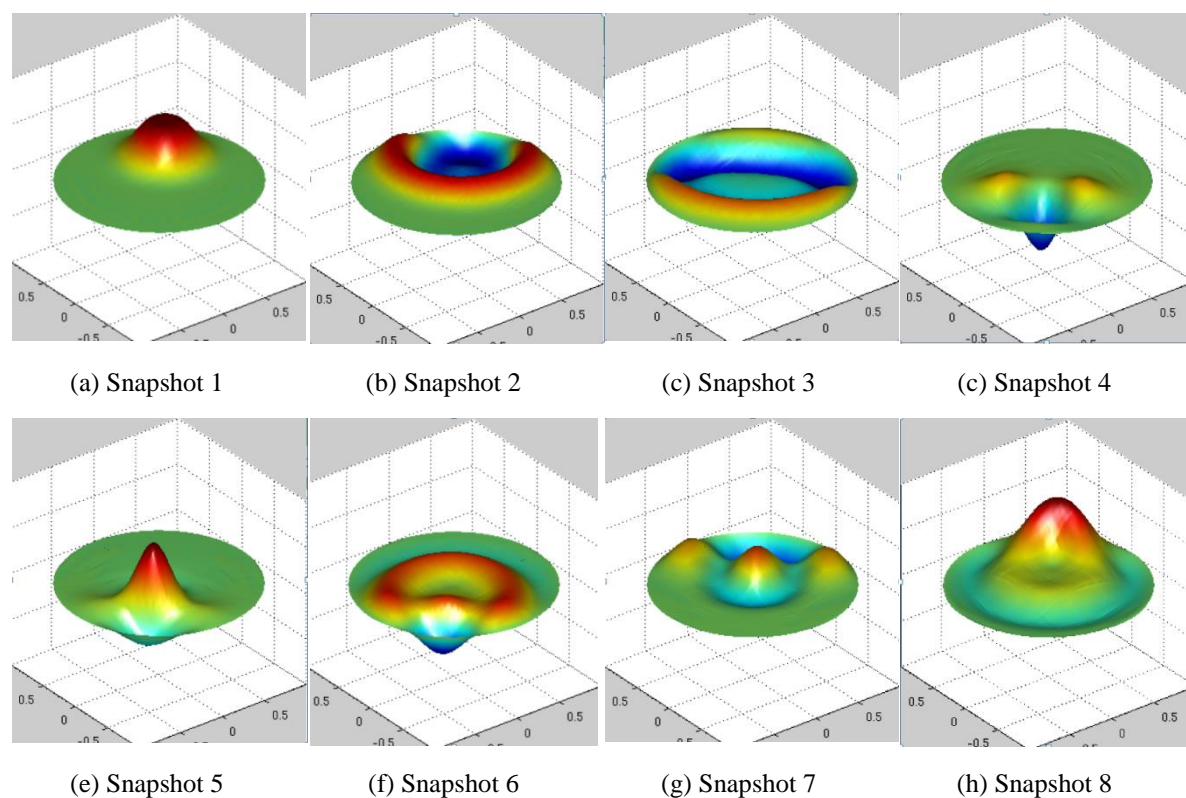
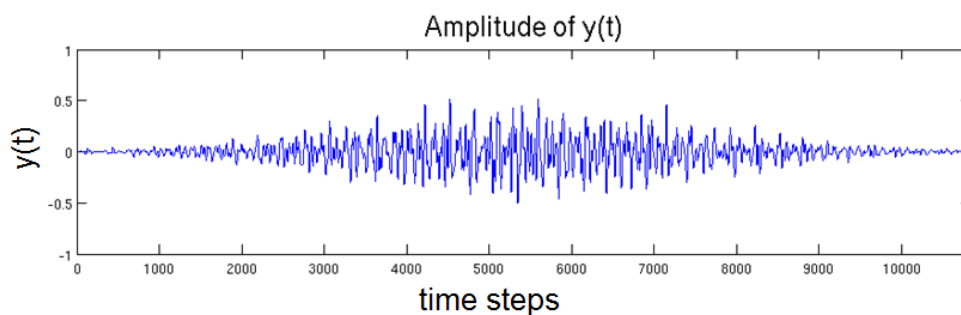
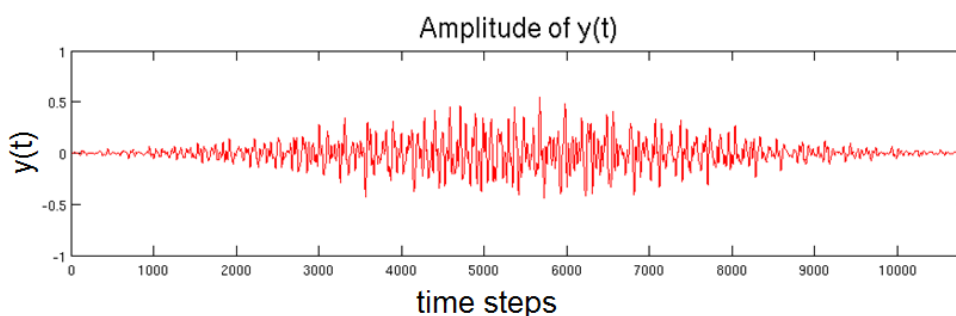


Figure 36: An off-center electric pulse is excited in a circular domain (a)-(h). A very long transient analysis is carried out by the DG method with central flux on fifth order elements. Fifth order is applied for the field as well as the boundary approximation.

Very long time signals of the field strength are recorded with an electric probe at each time step. Therefore, field variations represent the process of wave propagation and reflection in the domain. The signal is multiplied with a Gaussian window function in time domain [86]. In the end, the Fourier transformation of the time signal provides a spectrum that shows the resonant frequencies of the given geometry.



(a) Time signals obtained with straight-edged elements.



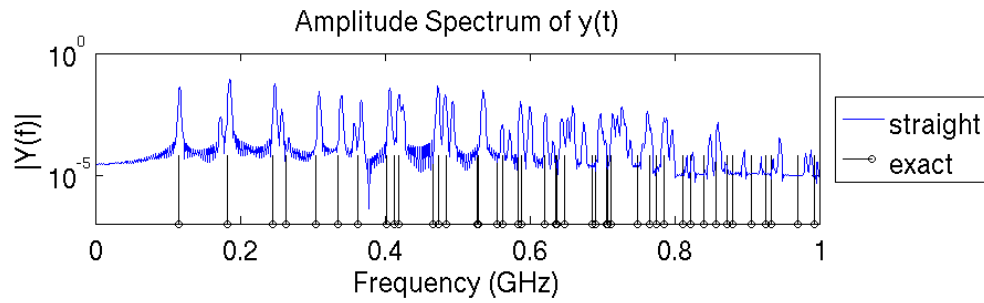
(b) Time signals obtained with curved elements.

Figure 37: Time signals obtained by the central flux DGM with 32 elements of 3rd order.

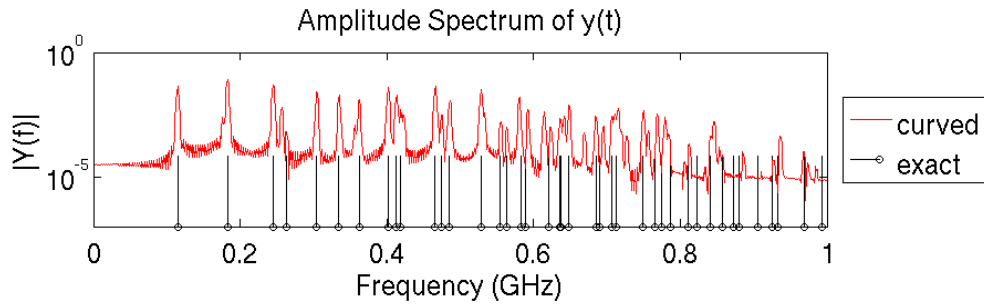
A validation of the numerical results can be made with a spectrum analysis. After performing the transient simulation, electric signals detected by the probe are recorded in time-domain as Figure 37 shows. An eyeball inspection of the two signals already shows clear differences as a result of the different quality of the boundary representation. The frequency spectrum has to be obtained by means of Fourier transform.

By solving the Helmholtz's equations in the frequency domain, we have already shown in equation (4-19) the analytical eigenvalues of the circular cavity. The exact eigenvalues are plotted with black stems in Figure 38 as reference results.

The spectra shown in Figure 38 are both obtained using 32 elements of 3rd order. However, the blue and red spectra are calculated with straight-edge elements and curved elements, respectively. Only the first eigenvalue of the blue one agrees very well with the exact value, all others are either shifted or split.



(a) Spectrum obtained with straight-edged elements.



(b) Spectrum obtained with curved elements.

Figure 38: Frequency spectrum obtained by the central flux DGM with 32 elements of 3rd order.

The reason for the shift and mismatch of spectrum can be interpreted from the second example in Figure 33, specifically from the oscillations of the fields in the corners. More accurately, the volume integration and surface integrations of the straight-edge elements do not represent the geometry accurately enough. Meanwhile, the red spectrum with curved elements agrees well with the exact solutions up to 0.6 GHz.

Increasing the polynomial degree to 6th order, the frequency spectrum with the red solid lines is shown in Figure 39. Resonance frequencies of the spectrum agree with the analytical solutions with the black stems. Almost all the eigenvalues up to 1 GHz are captured by the red spectrum. As a comparison, the previously obtained spectrum obtained with 3rd order approximation is indicated with the blue dashed line. Some of the resonant frequencies above 0.6 GHz are shifted from their respective exact values. Considering that only 32 elements are used, this DG scheme is highly accurate and very competitive with regard to eigenvalue solutions.

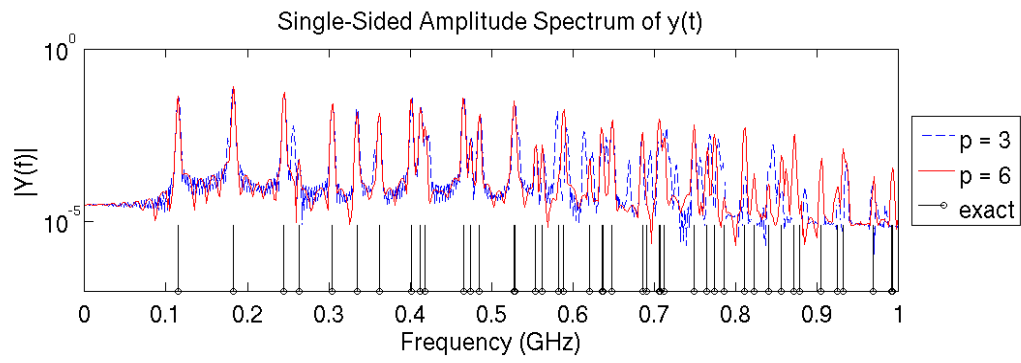


Figure 39: Frequency spectrum obtained with 32 elements using 3rd order polynomials (blue dashed line) and with 6th order polynomial (red solid line) and the corresponding exact eigenvalues solutions (black stems).

5. Summary and Outlook

5.1. Summary

In this work, boundary fitting techniques based on quadrilateral/hexahedral meshes have been designed for the high order Discontinuous Galerkin Method. The fitting procedures have been implemented in two levels: h -fitting, i.e. matching object and domain boundaries with the element vertices (see Section 2.1), and p -fitting, i.e. curving edges and faces of the elements appropriately (see Section 2.3). The mesh generation steps and validations include:

- After introducing element types and mesh topologies, a grid-based approach (see Section 2.1.3) for the geometry discretization has been chosen, which yields quasi-structured meshes. On the h -fitting level, the buffer layer scheme with a distance function provides a possibility for generating meshes, which are piecewise conforming with curved geometries. It was shown that the application of both, the buffer layer technique and the distance function largely facilitate the embedding of features into body-fitting quadrilateral/hexahedral meshes.
- On the p -fitting level, the TFI technique has been applied for obtaining high order curved elements, where TFI interpolates points on possibly curved boundaries within each element of the mesh. It has been demonstrated that curved elements are an important factor for improving the mesh quality in order to obtain accurate solutions with the DGM while avoiding an excessive number of elements.
- The mesh scheme was validated through examples with curved boundaries given in both implicit and explicit form. It also was extended into three dimensions with a variety of mesh examples. It has been shown that the geometric error reduces when the boundaries are described with high order accuracy. In particular, it has been shown that the errors in the computation of area and volume measures in two- and three-dimensional curved domains converge at a high rate when high order quadrature rules are applied with the curved element meshes.

The DGM is a very general and highly accurate numerical method. Considering the selected problem, one is free to choose the class of the basis functions, the time integrator as well as the type of the numerical flux. The following properties and applications of the DGM have been shown:

- A hierarchical modal basis of Legendre polynomials was employed with the DG method. The resulting block diagonal mass matrix was shown to yield an efficient scheme that is explicit in time.
- A discretization of Maxwell's equations has been performed with the central flux DGM for electromagnetic simulations. The advantage of curved elements has been shown by comparing the reflections of the waves at polygonal and circular boundaries, respectively.
- The new mesh scheme has been applied to resonant cavity problems. High order convergence results have been demonstrated in both square and cylinder cavities. Particularly, it has been shown that optimal convergence rates are obtained only if high order curved elements are employed. In this regard, boundaries have been represented using non-polynomial rational curves.
- Additionally, transient simulations with the central flux DGM were carried out for computing eigenvalues in a circular domain. A wide band of eigenvalues in the frequency domain was obtained through Fourier analysis of the recorded time signals. The resulting spectra are clean, i.e. free of any spurious modes. All the resonant frequencies agree well with their respective analytical solutions.

5.2. Future Work

The main restriction to automatic all hexahedral mesh generation stems from the inconsistency between geometry and mesh topology. Recent research concentrates on embedding topological features of the geometry into mostly Cartesian meshes. The isomorphic relation between elements at boundaries and the background mesh (see Section 2.1.3) should be incorporated with the introduced grid-based method. This approach has recently been employed in other research groups as well and is becoming one of the most actively developed hexahedral mesh algorithms [21].

A number of open questions remain on how to associate local features like points, lines and surfaces of the geometry with vertices, edges and faces of the background mesh. Therefore, future work should include:

- Generalized methods for constructing buffer layers based on parameterized boundaries instead of the currently taken problem-specific approach.
- Since high order elements are computationally expensive, adaptive mesh refinement and coarsening should be employed. In this case the curvature of elements has to be adapted alongside with the element size and order. This requires a mesh density indicator built on the background mesh.

- Regarding problems in three-dimensional space, only geometries with implicit boundary definitions have been applied with the grid-based method. However, NURBS surfaces have to be included as a further improvement of the method for coping with real-world three-dimensional examples.

From a practical point of view, complex examples using the new mesh scheme with the DGM should be fully investigated in two- and three-dimensional space:

- The horn antenna shown in the end of Section 2.4 is one such example, where accurate results should be obtained with a relatively small number of high order curved elements.
- With the new mesh scheme, the method is a truly high order method in space using high order approximations of the electromagnetic fields as well as objects and boundaries. It is therefore a natural next step to address the subject of high order time integration for reducing the number of time steps.
- In order to take full benefit of the quadtree/octree refined meshes, hanging nodes should be incorporated in the implementation. This would allow for the highly localized application of curved elements while maintaining Cartesian elements throughout most of the mesh resulting in a highly efficient scheme. Then, applications such the two-dimensional airfoil [87] in Figure 40 could be tackled. A body-fitting mesh is shown in Figure 41.

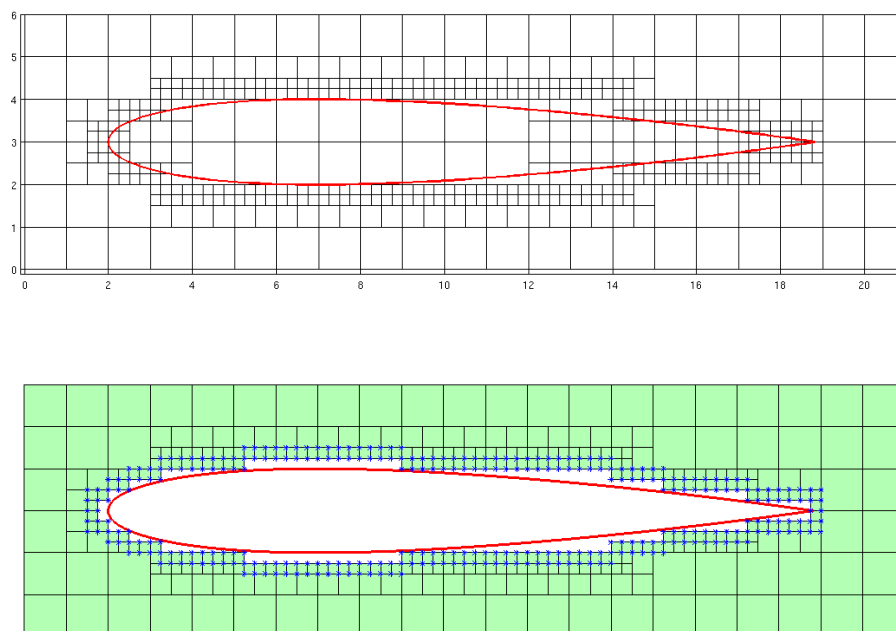


Figure 40: A quadtree mesh for a NACA 0012 airfoil. The upper picture shows the quadtree background mesh, while the lower one shows the selected elements in the outer domain of the airfoil.

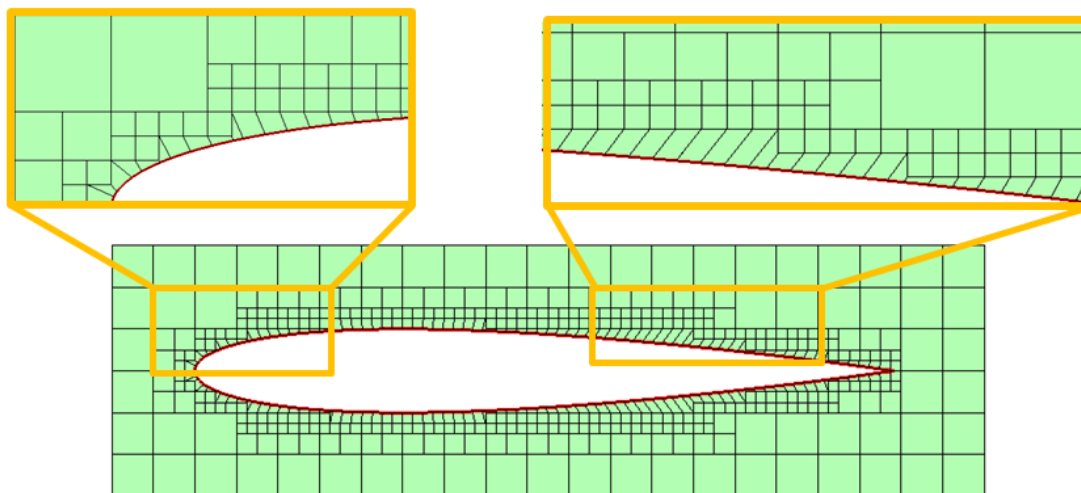


Figure 41: The selected part of the background mesh is combined with buffer layer mesh in order to obtain a body-fitting mesh for the NACA airfoil model.

Appendix A

Table A-1 Roots of Bessel functions (m-th root of the n-th order $J_n(x) = 0$) [83].

n \ m	1	2	3	4	5
0	2. 404825558	5. 52007811	8. 653727913	11. 79153444	14. 93091771
1	3. 83170597	7. 01558667	10. 17346814	13. 32369194	16. 47063005
2	5. 135622302	8. 41724414	11. 61984117	14. 79595178	17. 95981949
3	6. 380161896	9. 76102313	13. 01520072	16. 22346616	19. 40941523
4	7. 588342435	11. 06470949	14. 37253667	17. 61596605	20. 82693296
5	8. 771483816	12. 3386042	15. 70017408	18. 98013388	22. 2177999
6	9. 936109524	13. 58929017	17. 00381967	20. 32078921	23. 58608444
7	11. 08637002	14. 82126873	18. 28758283	21. 64154102	24. 93492789
8	12. 22509226	16. 03777419	19. 55453643	22. 94517313	26. 26681464
9	13. 35430048	17. 24122038	20. 80704779	24. 23388526	27. 58374896
10	14. 47550069	18. 43346367	22. 04698536	25. 50945055	28. 88737506

Table A-2 Roots of Derivatives of Bessel functions (m-th root of the n-th order $J'_n(y) = 0$) [83]

n \ m	1	2	3	4	5
0	3. 83170597	7. 01558667	10. 17346814	13. 32369194	16. 47063005
1	1. 841183781	5. 331442774	8. 536316366	11. 7060049	14. 86358863
2	3. 054236928	6. 706133194	9. 969467823	13. 17037086	16. 34752232
3	4. 201188941	8. 015236598	11. 34592431	14. 58584829	17. 78874787
4	5. 317553126	9. 282396285	12. 68190844	15. 96410704	19. 1960288
5	6. 415616376	10. 51986087	13. 98718863	17. 31284249	20. 57551452
6	7. 501266145	11. 73493595	15. 26818146	18. 63744301	21. 93171502
7	8. 57783649	12. 93238624	16. 52936588	19. 94185337	23. 26805293
8	9. 647421652	14. 11551891	17. 77401237	21. 22906262	24. 58719749
9	10. 71143397	15. 28673767	19. 00459354	22. 50139873	25. 89127728
10	11. 77087667	16. 44785275	20. 22303141	23. 76071586	27. 18202153

List of Figure

Figure 1: A one dimensional problem is solved by 4, 6 and 12 elements with 1st order polynomial basis.4

Figure 2: A one dimensional problem is solved by two elements with 1st, 2nd and 4th order polynomials.4

Figure 3: Exponential convergence in one dimension is depicted with the blue line for increasing the order, the green one for increasing the number of 1st order elements. The left side (a) plot shows the initial interpolation errors, while the right side (b) plot shows the errors measured after one period of oscillation.5

Figure 4: One quarter of a circular domain Ω (closed by a curved and two straight boundaries). Mesh (b) is obtained by h -refinement of mesh (a). Further r -refinement based on mesh (b) leads to a fitting mesh (c), where vertices of elements were relocated to the curved boundary.6

Figure 5: One quarter of a circular domain Ω (closed by a curved and two straight boundaries). Mesh (b) is obtained by p -refinement of mesh (a). Further r -refinement based on mesh (b) leads to a fitting mesh (c), where the interpolation points were rearranged on the curved boundary.6

Figure 6 : A square domain Ω (closed by straight boundaries) is a special case where the approximated domain Ω_h introduces no geometric error. The different meshes can be described in two ways: structured (a, b) and unstructured meshes (c, d), or quadrilateral (a, c) and triangular element meshes (b, d). 12

Figure 7: One quarter of a circular domain Ω (closed by a curved and two straight boundaries) is approximated using four mesh schemes: (a) Cartesian, (b) Cartesian with quadtree refinement and (c, d) unstructured meshes. The approximation to the curved boundary is composed by the union of boundary edges of the elements, which are either (a, b) piece-wise constant, or (c, d) piece-wise linear. 14

Figure 8 : A lake domain with curved boundary is discretized by a quadtree mesh. The small details of the boundary are captured by the refined mesh. 15

Figure 9 : Refinement templates and their refined mesh for one quarter of a circular domain before the selected elements (green color) are deformed. 17

Figure 10 : One quarter of a circular domain Ω (closed by a curved and two straight boundaries) is covered by quadrilateral meshes: (a) Cartesian mesh (b)

Template refined mesh (c) Unstructured refined mesh (d) Quasi-structured mesh.....	18
Figure 11 : Mesh generation of a moving curved boundary in a square domain. The quadtree and “Y-shaped” template refinements are carried out at the interface around the curved boundary.....	20
Figure 12 : A circular boundary $\partial\Omega$ is represented by implicit and explicit forms. (a) The implicit form uses a signed function $f(x,y)$ to tell whether a point is on the boundary, while (b) the explicit one relies on NURBS to parameterize the boundary where red nodes are the control points for the basis functions.....	21
Figure 13 : A circular domain Ω is tessellated with elements according to two mesh schemes: (c) twelve degenerate elements are produced near the boundary, while in (f) geometry-oriented and well-shaped elements are obtained.....	24
Figure 14 : Meshes are equipped with buffer layer elements. Two examples of implicit forms are considered: (a) an ellipse and (b) a coaxial cable.....	25
Figure 15 : Buffer layer elements and basis functions for an explicit form of a deformed circle.	25
Figure 16 : Arbitrary curved boundaries can be described using NURBS (a). During the meshing process first an initial staircase mesh is generated (b), the distance functions is computed (c) and then the buffer layer elements are inserted between the boundary and the interior staircase mesh.....	27
Figure 17 : Mesh example obtained by the buffer mesh scheme using NURBS, which employ piecewise linear basis functions.	28
Figure 18 : Mesh example obtained by the buffer mesh scheme using NURBS, which employ piecewise quadratic basis functions.	29
Figure 19 : A mapping function transforms a hexahedron from reference coordinates (ξ, η, ζ) to physical coordinates x, y, z with a curved surface $\{v_5 v_6 v_7 v_8\}$..	31
Figure 20 : A mapping function in transforms a quadrilateral from reference coordinates (ξ, η) to physical coordinates x, y with a curved edge $\{v_3 v_4\}$, which is parameterized by $x, y = F(\xi, 1)$ (a). A TFI implementation is demonstrated for a patch with uniform distributed points in (b) and (c).	33
Figure 21 : Numerical integration test of a circular domain area using the high order curved element mesh scheme with different quadrature orders.....	36
Figure 22 : Three-dimensional Gaussian distributions of electric field in (a) polyhedral mesh and in (b) polyhedral domains truncated from the sphere.....	37
Figure 23 : Examples of the generation of body-fitting meshes based on a structured interior mesh and its corresponding buffer layers elements in three dimensions.	38

- Figure 24 : Three-dimensional Gaussian distributions of electric field in (b) (e) spherical domains and in (c) (f) polyhedral domains truncated from the sphere.39
- Figure 25 : Numerical integration test of a sphere domain volume using the high order curved element mesh scheme with different quadrature orders.39
- Figure 26 : Meshing a circular horn antenna. Elements generated in the buffer layers are geometry-oriented, i.e. new elements (in green color) are inserted along the surfaces of the geometry.40
- Figure 27 Lagrange polynomial basis functions of order 7 (on the left hand side). Legendre polynomial basis functions of order 7 (on the left hand side).42
- Figure 28: Nodal basis functions with Lagrange polynomials from 1st to 3rd order. .45
- Figure 29: Hierarchical modal basis with Legendre polynomials up to 3rd order, where the lower order polynomials are all included in the higher order ones...45
- Figure 30: Discretization of a domain consisting of a quarter circle (closed by red arc and two boundaries), the discrete domains Ω_h with (a) rectangles, (b) quadrilaterals and (c) the approximation domain $\Omega_{h,p}$ with curved-edge quadrilaterals.....51
- Figure 31: A square domain is covered with (a) h-refined and (b) p-refined meshes, on which (c) (d) the initial electric field distributions are based. After discretization with the DG method errors as plotted in (e) (f) are obtained for 0th and 3rd order polynomials.60
- Figure 32: L_2 error of electric field with polynomial increasing from 1st to 5th order. The errors show $(p+1)$ convergence slopes in a logarithmic plot.61
- Figure 33: A circular domain is discretized with three meshes with interpolation points of 6th order (a)-(c). The computation is performed by the central flux DG method with 6th order polynomial basis. For the polygonal and smooth domains excitation occurs in the center. The resulting electric fields before and after reflection are depicted in (d)-(f) and (g)-(i). Only for curved elements, the wave remains cylindrical after reflection.64
- Figure 34: Snapshots of the electric field distribution of the TM₃₁ mode in half a period of oscillation. The computation is performed by the central flux DG method with 56 elements of 6th order polynomial basis.65
- Figure 35: A resonant TM₃₁ mode in the circular cavity is calculated on (a) three sets of body-fitting meshes. The z-component of the electric field (b) is plotted with the analytical solutions. The numerical solutions obtained by the DG method with upwind flux (c) shows $(p+1)$ convergence orders.67
- Figure 36: An off-center electric pulse is excited in a circular domain (a)-(h). A very long transient analysis is carried out by the DG method with central flux on

fifth order elements. Fifth order is applied for the field as well as the boundary approximation.	71
Figure 37: Time signals obtained by the central flux DGM with 32 elements of 3 rd order.	72
Figure 38: Frequency spectrum obtained by the central flux DGM with 32 elements of 3 rd order.....	73
Figure 39: Frequency spectrum obtained with 32 elements using 3 rd order polynomials (blue dashed line) and with 6 th order polynomial (red solid line) and the corresponding exact eigenvalues solutions (black stems).....	74
Figure 40: A quadtree mesh for a NACA 0012 airfoil. The upper picture shows the quadtree background mesh, while the lower one shows the selected elements in the outer domain of the airfoil.	77
Figure 41: The selected part of the background mesh is combined with buffer layer mesh in order to obtain a body-fitting mesh for the NACA airfoil model.	78

Bibliography

1. W. H. Reed and T. R. Hill, “Triangular mesh methods for the neutron transport equation”, *Los Alamos Report LA-UR-73-479*, 1973.
2. A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 0003 ed. Artech House Inc, 2005.
3. J. S. Hesthaven, “High-order accurate methods in time-domain computational electromagnetics: A review”, *Advances in Imaging and Electron Physics*, vol 127, pp 59–123, 2003.
4. B. Guo and I. Babuvska, “The hp version of the finite element method”, *Computational Mechanics*, vol. 1, no. 1, pp. 21–41, 1986.
5. F. Bassi and S. Rebay, “High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations”, *J. Comput. Phys.* 138(2), 251–285, 1997.
6. J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Verlag, 2007.
7. L. A. Piegl and W. Tiller, *The Nurbs Book*. Springer, 1997.
8. D. Wirasaet, S. Tanaka, E. J. Kubatko, J. J. Westerink, and C. Dawson, “A performance comparison of nodal discontinuous Galerkin methods on triangles and quadrilaterals”, *Int. J. Numer. Math. Fl.* 64(10–12), 1336–1362, 2010.
9. T. E. Peterson, “A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation,” *SIAM Journal on Numerical Analysis*, vol. 28, no. 1, pp. 133–140, 1991.
10. J. Qiu, B. C. Khoo, and C. W. Shu, “A numerical study for the performance of the Runge–Kutta discontinuous Galerkin method based on different numerical fluxes,” *Journal of Computational Physics*, vol. 212, no. 2, pp. 540–565, 2006.
11. P. Rasetarinera, M. Hussaini, and F. Hu, “Some remarks on the accuracy of a discontinuous Galerkin method,” *Lecture Notes in Computational Science and Engineering*, vol. 11, pp. 407–412, 2000.
12. K. Yee, “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media”, *Antennas and Propagation, IEEE Transactions on*, vol 14, no 3, pp 302–307, 1966.
13. O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals*, vol 1. Butterworth-Heinemann, 2005.
14. A. Bossavit, *Computational electromagnetism: variational formulations, complementarity, edge elements*, vol 2. Academic Press, 1997.
15. R. J. LeVeque, *Finite volume methods for hyperbolic problems*, vol 31. Cambridge university press, 2002.
16. C. Fumeaux, D. Baumann, and R. Vahldieck, “Finite-volume time-domain analysis of a cavity-backed Archimedean spiral antenna”, *Antennas and Propagation, IEEE Transactions on*, vol 54, no 3, pp 844–851, 2006.
17. P. LeSaint, “On a finite element method for solving the neutron transport equation”, Academic Press, 1974.

18. B. Cockburn, G. E. Karniadakis, and C. W. Shu, "The development of discontinuous Galerkin methods," *UMSI research report/University of Minnesota (Minneapolis, Mn). Supercomputer institute*, vol. 99, p. 220.
19. B. Cockburn and C. W. Shu, "The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems", *Journal of Computational Physics*, vol 141, no 2, pp 199–224, 1998.
20. "Mesh generation", *Wikipedia, the free encyclopedia*. 12-Nov-2012.
21. S. J. Owen and J. F. Shepherd, "Embedding Features in a Cartesian Grid", *Proceedings of the 18th International Meshing Roundtable*, pp. 117–138, 2009.
22. P. O. Persson, "Mesh size functions for implicit geometries and PDE-based gradient limiting", *Engineering with Computers*, vol. 22, no. 2, pp. 95–109, 2006.
23. K. Shimada, J. H. Liao, and T. Itoh, "Quadrilateral meshing with directionality control through the packing of square cells", in *Proceedings of 7th International Meshing Roundtable*, pp. 61–75, 1998.
24. X. Roca, E. Ruiz-Gironés, and J. Sarrate, "Receding front method: a new approach applied to generate hexahedral meshes of outer domains", *Proceedings of the 19th International Meshing Roundtable*, pp. 209–225, 2010.
25. D.-W. Sun, *Computational Fluid Dynamics in Food Processing*. CRC Press, 2007.
26. D. Ryppl, *Sequential and parallel generation of unstructured 3D meshes*. Czech Technical University, 1998.
27. P. O. Persson and G. Strang, "A simple mesh generator in MATLAB", *SIAM review*, vol. 46, no. 2, pp. 329–345, 2004.
28. A. Loseille and R. Löhner, "Robust Boundary Layer Mesh Generation", *Proceedings of the 21st International Meshing Roundtable*, pp 493–511, 2012.
29. R. Löhner and P. Parikh, "Generation of three-dimensional unstructured grids by the advancing-front method", *International Journal for Numerical Methods in Fluids*, vol 8, no 10, pp 1135–1149, 1988.
30. R. Löhner, "Three-dimensional grid generation by the advancing-front method", *Numerical methods in laminar and turbulent flow*, 1987.
31. J. Schöberl, "NETGEN An advancing front 2D/3D-mesh generator based on abstract rules", *Computing and visualization in science*, vol 1, no 1, pp 41–52, 1997.
32. Y. Ito, A. M. Shih, and B. K. Soni, "Reliable isotropic tetrahedral mesh generation based on an advancing front method", *Proceedings of the 13th International Meshing Roundtable, Williamsburg, VA*, pp 95–105, 2004.
33. P. L. George and E. Seveno, "The advancing-front mesh generation method revisited", *International Journal for Numerical Methods in Engineering*, vol 37, no 21, pp 3605–3619, 1994.
34. N. P. Weatherill and O. Hassan, "Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints", *International Journal for Numerical Methods in Engineering*, vol 37, no 12, pp 2005–2039, 1994.
35. D. T. Lee and A. K. Lin, "Generalized Delaunay triangulation for planar graphs", *Discrete & Computational Geometry*, vol 1, no 1, pp 201–217, 1986.
36. D. A. Field, "Laplacian smoothing and Delaunay triangulations", *Communications in applied numerical methods*, vol 4, no 6, pp 709–712, 1988.
37. J. D. Boissonnat and B. Geiger, "Three dimensional reconstruction of complex shapes based on the Delaunay triangulation", 1992.
38. D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation", *International Journal of Parallel Programming*, vol 9, no 3, pp 219–242, 1980.

39. W. J. Schroeder and M. S. Shephard, "A combined octree/delaunay method for fully automatic 3-D mesh generation", *International Journal for Numerical Methods in Engineering*, vol 29, no 1, pp 37–55, 1990.
40. H. Borouchaki, P. Laug, and P. L. George, "Parametric surface meshing using a combined advancing-front generalized Delaunay approach", *International Journal for Numerical Methods in Engineering*, vol 49, no 1–2, pp 233–259, 2000.
41. J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, J. D. Teresco, and L. H. Ziantz, "Adaptive local refinement with octree load balancing for the parallel solution of three-dimensional conservation laws", *Journal of Parallel and Distributed Computing*, vol 47, no 2, pp 139–152, 1997.
42. L. Horesh and E. Haber, "A Second Order Discretization of Maxwell's Equations in the Quasi-Static Regime on OcTree Grids", *SIAM Journal on Scientific Computing*, vol 33, no 5, pp 2805–2822, 2011.
43. S.M. Schnepf and T. Weiland, "Efficient Large Scale Electromagnetics Simulations Using Dynamically Adapted Meshes with the Discontinuous Galerkin Method", *J. Comput. Appl. Math.*, vol 236, pp 4909–4924, 2012
44. G. P. Nikishkov, "Finite element algorithm with adaptive quadtree-octree mesh refinement", *ANZIAM Journal*, vol 46, pp C15–C28, 2005.
45. R. Schneiders, "Algorithms for quadrilateral and hexahedral mesh generation", *Proceedings of the VKI Lecture Series on Computational Fluid Dynamics*, 2000.
46. Y. Zhang and C. Bajaj, "Adaptive and quality quadrilateral/hexahedral meshing from volumetric data", *Computer methods in applied mechanics and engineering*, vol 195, no 9, pp 942–960, 2006.
47. S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal, "Q-Morph: an indirect approach to advancing front quad meshing", *International Journal for Numerical Methods in Engineering*, vol 44, no 9, pp 1317–1340, 1999.
48. S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal, "Advancing front quadrilateral meshing using triangle transformations", in *7th International Meshing Roundtable*, 1998, pp 409–428.
49. S. J. Owen, "A survey of unstructured mesh generation technology", in *7th International Meshing Roundtable*, 1998, vol 3.
50. P. Lavoie, "An introduction to NURBS", Technical report, Ottawa university, 1999. <http://yukon.genie.uottawa.ca/~lavoie/software/nurbs>, 1999.
51. <http://mathworld.wolfram.com/BezierCurve.html>
52. <http://www.rose-hulman.edu/~finn/CCLI/notes.htm>
53. C. Blanc and C. Schlick, "More Accurate Representation of Conics by NURBS", *Relation*, vol 10, no 1.44, p 5770, 1995.
54. "Non-uniform rational B-spline", *Wikipedia, the free encyclopedia*. 12-Nov-2012.
55. W. J. Gordon and C. A. Hall, "Transfinite element methods: blending-function interpolation over arbitrary curved element domains", *Numerische Mathematik*, 21(2), 109–129, 1973
56. P. G. Ciarlet, *The finite element method for elliptic problems*, vol. 4. North Holland, 1978.
57. S. Schnepf, E. Gjonaj, and T. Weiland, "A time-adaptive mesh approach for the self-consistent simulation of particle beams", in *Proc. 9th Int. Comp. Accel. Phys. Conf.*, 2006, pp 132–135.
58. S. Schnepf, E. Gjonaj, and T. Weiland, "Development of a Self-Consistent Particle-In-Cell (PIC) Code using a Time-Adaptive Mesh Technique", in *Proc. 10th European Particle Accelerator Conference (EPAC)*, 2006, pp 2182–2184.

59. S. Schnepf, “Space-Time Adaptive Methods for Beam Dynamics Simulations”, TU Darmstadt, 2009.
60. S. M. Schnepf and T. Weiland, “The Discontinuous Galerkin Method on Dynamical hp-Meshes” presented at the SCEE 2012, Zurich.
61. P. O. Persson, “Mesh generation for implicit geometries”, *Citeseer*, 2004.
62. I. Ergatoudis, B. M. Irons, and O. C. Zienkiewicz, “Curved, isoparametric, ‘quadrilateral’ elements for finite element analysis”, *International Journal of Solids and Structures*, vol. 4, no. 1, pp. 31–42, 1968.
63. T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement”, *Computer methods in applied mechanics and engineering*, vol 194, no 39, pp 4135–4195, 2005.
64. A. Buffa, G. Sangalli, and R. Vázquez, “Isogeometric analysis in electromagnetics: B-splines approximation”, *Computer Methods in Applied Mechanics and Engineering*, vol 199, no 17, pp 1143–1152, 2010.
65. R. Sevilla, S. Fernández-Méndez, and A. Huerta, “NURBS-enhanced finite element method (NEFEM)”, *International Journal for Numerical Methods in Engineering*, vol 76, no 1, pp 56–83, 2008.
66. J. Stoer and R. Bulirsch, *Introduction to numerical analysis*. Springer-Verlag, 1980.
67. J. C. Nédélec, “Mixed finite elements in \mathbb{R}^3 ”, *Numerische Mathematik*, vol. 35, no. 3, pp. 315–341, 1980.
68. M. Ainsworth, “Dispersive properties of high-order Nédélec/edge element approximation of the time-harmonic Maxwell equations”, *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 362, no. 1816, pp. 471–491, 2004.
69. “Gaussian quadrature”, *Wikipedia, the free encyclopedia*. 18-Nov-2012.
70. E. W. Weisstein, “Lobatto Quadrature -- from Wolfram MathWorld”. [Online]. Available: <http://mathworld.wolfram.com/LobattoQuadrature.html>. [Accessed: 22-Nov-2012].
71. “Nodes and Weights of Gauss-Lobatto - High accuracy calculation for life or science.”, *High accuracy calculation for life or science*. [Online]. Available: <http://keisan.casio.com/>. [Accessed: 22-Nov-2012].
72. E. Makris, P. Neofytou, S. Tsangaris, and C. Housiadas, “A novel method for the generation of multi-block computational structured grids from medical imaging of arterial bifurcations”, *Medical Engineering & Physics*, vol 34, no 8, pp 1157–1166, 2012.
73. L. Antiga, J. Peiró, and D. A. Steinman, “From image data to computational domains”, in *Cardiovascular Mathematics*, L. Formaggia, A. Quarteroni, and A. Veneziani, Eds Springer Milan, 2009, pp 123–175.
74. E. Gjonaj, T. Lau, S. Schnepf, F. Wolfheimer, and T. Weiland, “Accurate modelling of charged particle beams in linear accelerators”, *New Journal of Physics*, vol 8, no 11, p 285, 2006.
75. B. Cockburn and C. W. Shu, “Runge–Kutta discontinuous Galerkin methods for convection-dominated problems”, *Journal of Scientific Computing*, vol 16, no 3, pp 173–261, 2001.
76. D. White, A. Fisher, M. Stowell, J. Koning, R. Rieben, N. Champagne, and N. Madsen, *Higher-order mixed finite element methods for time domain electromagnetics*. United States. Department of Energy, 2004.
77. J. W. Thomas, *Numerical partial differential equations: finite difference methods*, vol 1. Springer, 1995.

78. R. Rieben, D. White, and G. Rodrigue, “High-order symplectic integration methods for finite element solutions to time dependent Maxwell equations”, *Antennas and Propagation, IEEE Transactions on*, vol 52, no 8, pp 2190–2195, 2004.
79. J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral methods for time-dependent problems*, vol. 21. Cambridge University Press, 2007.
80. K. Zhang, D. Li, K. Chang, K. Zhang, and D. Li, *Electromagnetic theory for microwaves and optoelectronics*. Springer, 2008.
81. Z. Lou and J. M. Jin, “An accurate waveguide port boundary condition for the time-domain finite-element method”, *Microwave Theory and Techniques, IEEE Transactions on*, vol 53, no 9, pp 3014–3023, 2005.
82. D. H. Choi and W. J. R. Hoefer, “The finite-difference-time-domain method and its application to eigenvalue problems”, *Microwave Theory and Techniques, IEEE Transactions on*, vol 34, no 12, pp 1464–1470, 1986.
83. <http://wwwal.kuicr.kyoto-u.ac.jp/www/accelerator/a4/besselroot.htmlx>
84. K. D. Devine and J. E. Flaherty, “Parallel adaptive hp-refinement techniques for conservation laws”, *Applied Numerical Mathematics*, vol 20, no 4, pp 367–386, 1996.
85. P. Houston and E. Süli, “A note on the design of hp-adaptive finite element methods for elliptic partial differential equations”, *Computer methods in applied mechanics and engineering*, vol 194, no 2, pp 229–243, 2005.
86. T. Banova, banova@temf.tu-darmstadt.de
87. http://turbmodels.larc.nasa.gov/naca0012_val.html

Acknowledgements

At the end of this dissertation, I would like to express my gratitude to all the people that have helped me in various ways.

I would like to thank Prof. Dr.-Ing. Thomas Weiland for providing this great opportunity of scientific research in TU Darmstadt. Without his support and supervision, it would be impossible for me to complete this Ph.D. promotion.

I would also like to thank Prof. Dr.-Ing. Min Zhang for guiding me into current research area. Moreover, his friendly family members have shared with me countless of weekends and festivals in Germany.

Special thanks are given to my tutor Dr. Schnepf, who has guided my work and has carefully corrected this manuscript. We had worked together in Darmstadt for three years before he went to ETH Zurich in Swiss. His patience and devoted working spirit encourages me a lot.

During the research activities at TEMF and Graduate School of Computational Engineering, all my colleagues and management staff members are always ready to help regardless of their own work at hand. Their generous help are forever written in my memory

I thank previous Ph.D. students Yu Du, Jing Gao and Stephen as well as CEM members Martin, Fritz and Alexander. Their diligent working attitudes inspire me into the future career.

I also thank current colleagues Xingyuan, Haitao, Jane, Debora, Nicklas,....., from CE and Klaus, Xiao, Todorka, Annette, Fedor, Fatih, Olha,....., from TEMF. They have been keeping an ideal atmosphere for academic research.

I am thankful to many groups and social activities that bring fresh air into the tough research work. These include: cinema event, lunch group, Christmas party and KWT event from TEMF, evening event and marathon relay from CE, and sports group from FNB. This part of experience is equally important during my studying abroad.

Finally, I want to express the appreciation to my mother and father, whose constant love supports me pursuing my dreams.

Thank you all for making my study in Darmstadt a pleasant life journey!

Curriculum Vitae



Jian Cui

4 th August 1982	Born in Fushun, Liaoning Province, China
1989-1991	Gebu No.2 primary school in Fushun, China
1991-1995	Yuhong primary school in Fushun, China
1995-1998	Dongzhou junior middle school in Fushun, China
1998-2001	Fushun No.10 senior middle school in Fushun, China
2001-2005	Undergraduate study of Electrical Engineering at Kunming University of Science and Technology, Kunming, China
2005-2008	Graduate study of Electrical Engineering at Tongji University, Shanghai, China
2009-2012	Ph.D. study at Computational Engineering Graduate School, Darmstadt University of Technology, Darmstadt, Germany

