A boundary conformal Discontinuous Galerkin method for electromagnetic field problems on Cartesian grids

Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von Dipl.-Tech.-Math. Annette Fröhlcke aus Bietigheim-Bissingen
November 2013 — Darmstadt — D 17

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Elektrotechnik und Informationstechnik
Institut für Theorie Elektromagnetischer Felder (TEMF)

A boundary conformal Discontinuous Galerkin method for electromagnetic field problems on Cartesian grids

Genehmigte Dissertation von Dipl.-Tech.-Math. Annette Fröhlcke aus Bietigheim-Bissingen

1. Gutachten: Prof. Dr.-Ing. Thomas Weiland
2. Gutachten: Prof. Dr. rer. nat. Michael Schäfer

Tag der Einreichung: 16.05.2013
Tag der Prüfung: 16.09.2013

Darmstadt — D 17

Abstract

The thesis presents a novel numerical method based on the high order Discontinuous Galerkin (DG) method for three dimensional electrostatic and electro-quasistatic field problems where materials are of very complex shape and may move over time. A well-known example is water droplets oscillating on the surface of high voltage power transmission line insulators. The electric field at the surface of the insulator causes the oscillation of the water droplets. The oscillation, in turn, triggers partial discharges which have damaging effects on the polymer insulation layers of high voltage insulators.

The simulation of such phenomena is highly complex from an electromagnetic point of view. Most numerical methods which are applied to such field problems use conforming meshes where the elements are fitted exactly to the material geometry. This implies that the element interfaces conform to the material boundaries or material interfaces. In general, the generation of conforming meshes is computationally expensive. Furthermore, when dealing with materials that move over time, conforming meshes need to be adapted to the changing material geometry at each point in time.

To avoid the often computationally costly generation and adaption step of conforming meshes, the numerical method proposed in this thesis operates on a single fixed structured Cartesian mesh. First, field problems with non-moving materials are considered. To obtain accurate simulation results on field problems with complex-shaped materials, an additional approach, namely the *cut-cell discretization approach*, is applied. The *cut-cell discretization approach* subdivides the elements at material boundaries or interfaces into smaller sub-elements which are referred to as cut-cells. The approach is embedded into the Discontinuous Galerkin (DG) method for standard Cartesian meshes since the DG method allows for high order approximations and offers a great flexibility for additional approaches. Since the mesh is not fitted to the material geometry, geometrically small cut-cells might emerge. Therefore, two supplementary approaches, the *adaptive approximation order method* and the *cell merging method* are proposed which enable an accurate approximation even on geometrically small cut-cells. Furthermore, a DG hybridization is presented which lowers the number of degrees of freedom in domains where the high number of DG degrees of freedom is not necessary to obtain accurate results. The numerical method comprising all above mentioned approaches is labelled as *boundary conformal DG* (BCDG) method.

In a second step, the BCDG method is extended to field problems where materials move over time. We refer to this approach as *extension of the BCDG* (EBCDG) method. The EBCDG method adapts to the moving materials by recalculating only the cut-cells at each point in time while the underlying Cartesian grid is kept fixed. Therefore, no computationally expensive mesh adaption or mesh generation steps are needed.

The BCDG and the EBCDG method are applied to numerical examples of electrostatic (ES) and electro-quasistatic (EQS) field problems. First, numerical results of the BCDG method on a verification example of a cylindrical capacitor filled with two dielectric layers are shown. A convergence study and a comparison study illustrate the high accuracy of the BCDG method with respect to the number of degrees of freedom. Finally, the EBCDG method is applied to an example of a water droplet oscillating artificially on the insulation layer of a high voltage insulator. A convergence study demonstrates that even on a coarse mesh a high resolution of the potential and electric field solution can be achieved.

Kurzfassung

In dieser Arbeit wird eine neue numerische Methode vorgestellt, die auf der *Discontinuous Galerkin* (DG) Methode basiert und für dreidimensionale elektrostatische und elektroquasistatische Feldprobleme mit komplex geformten Materialgeometrien entwickelt wurde. Insbesondere können mit dieser Methode bewegte Materialränder behandelt werden. Ein bekanntes Anwendungsbeispiel ist die Simulation singulärer Wassertropfen auf Hochspannungsisolatoren, die sich durch Niederschlag oder Kondensation auf den polymeren Isolierstoffoberflächen der Hochspannungsisolatoren sammeln. Diese singulären Tropfen werden aufgrund des anliegenden elektrischen Feldes zu Oszillationen angeregt. Die oszillierende Wirkung des elektrischen Feldes auf die Wassertropfen bewirkt wiederum, dass Teil- bzw. Mikroentladungen auftreten, die die Isolierstoffoberfläche schädigen. Die numerische Simulation der Tropfenbewegung auf Hochspannungsisolatoren ist sehr komplex. Die meisten Simulationsverfahren für diese Art von Feldproblemen verwenden Rechengitter, deren Elementränder exakt mit den Materialrändern übereinstimmen. Allerdings sind diese Gitter im Vergleich zu strukturierten Gittern oft nur mit einem erheblich höheren Zeitaufwand zu generieren. Zudem müssen diese Gitter bei Simulationen mit bewegten Materialrändern in jedem Zeitschritt an die bewegte Materialgeometrie angepasst werden.

Die numerische Methode, die in dieser Arbeit vorgestellt wird, vermeidet den zeitaufwendigen Gittergenerierungs- bzw. Gitteranpassungsschritt, indem ein vorgegebenes strukturiertes kartesisches Gitter verwendet wird. Um auf kartesischen Gittern eine ausreichende Genauigkeit für Feldprobleme mit komplex geformten Materialgeometrien erhalten zu können, wird eine Methode eingeführt, die die Materialränder in der Diskretisierung exakt berücksichtigt. Diese sogenannte *Cut-Cell Diskretisierungsmethode* schneidet die Elemente an den Materialrändern in Unterelemente, die sogenannten *Cut-Cells*. Die *Cut-Cell Diskretisierungsmethode* wird in die DG Methode eingebettet, weil diese eine Verwendung hoher Approximationsordnungen ermöglicht. Da die DG Methode aufgrund ihrer methodischen Struktur eine große Flexibilität für zusätzliche Methoden bietet, ist die Einbettung auf natürliche Weise möglich.

Da das kartesische Gitter nicht mit den Materialrändern übereinstimmt, können beim Schneiden der Elemente geometrisch sehr kleine *Cut-Cells* entstehen, die die Konditionszahl der Problemstellung verschlechtern. Aus diesem Grund werden in dieser Arbeit zwei weitere Methoden vorgestellt, die *Adaptive Approximation Order* Methode und die *Cell Merging* Methode, die es ermöglichen, selbst bei sehr kleinen Cut-Cells genaue Simulationsergebnisse zu erzielen. Zudem wird eine *DG Hybridisierungsmethode* eingeführt, die die Freiheitsgrade des Verfahrens in bestimmten Gebieten herabsetzt, ohne zu einem Genauigkeitsverlust zu führen. Die Methode, bestehend aus allen oben aufgelisteten Verfahren, wird als *Boundary Conformal DG* Methode bzw. BCDG Methode bezeichnet.

Eine Erweiterung der BCDG Methode auf bewegte Materialien, die EBCDG Methode, wird im zweiten Teil der Arbeit vorgestellt. Die bewegten Materialränder werden in dieser Methode durch eine wiederholte Neuberechnung der *Cut-Cells* zu jedem Zeitschritt berücksichtigt. Eine zeitaufwendige erneute Gittergenerierung oder eine Gitteradaption ist daher bei dieser Methode nicht notwendig.

Im letzen Abschnitt dieser Arbeit werden die numerischen Simulationergebnisse der BCDG und der EBCDG Methode für verschiedene elektrostatische und elektroquasistatische Feldprobleme vorgestellt. Eine Konvergenzanalyse und eine Vergleichsanalyse für das Verifikationsbeispiel eines Zylinderkondensators mit zwei dielektrischen Materialen zeigen die hohe Genauigkeit der BCDG Methode auf. Abschließend werden die Simulationsergebnisse für das Anwendungsbeispiel eines Wassertropfens auf einer Isolierstoffoberfläche vorgestellt, bei der eine künstliche Tropfenbewegung betrachtet wird. Die Simulationsergebnisse belegen, dass mit der EBCDG Methode selbst auf Gittern mit wenig Elementen eine sehr hohe Auflösung des elektrischen Feldes und des Potentials erzielt werden kann.

Contents

1 Introduction

1.1 Motivation

Numerical simulations in the field of electromagnetics are becoming ever more important in academia and industry alike. On the one hand, this is made possible by the growing computing power of modern computers. On the other hand, more efficient and more accurate numerical methods have been developed in the last decades. Both developments enable the simulation of increasingly complex electromagnetic phenomena. This includes field problems where materials are either of complex shape or move over time. Such problems are the focus of this thesis. A well-known example is water droplets oscillating on the surface of high voltage power transmission line insulators. The electric field on the insulation layer causes an oscillation of the water droplets. This oscillation, in turn, triggers partial discharges which damage the insulation layer material [49]. Other examples with materials of complex shape motivating this work include biomedical applications where organs or blood vessels have to be simulated. When solving such problems, most often standard numerical discretization methods such as the Finite Difference Time Domain (FDTD) method [92], the Finite Integration Technique (FIT) [91], the Finite Element (FE) method [60] and the Discontinuous Galerkin (DG) [38] method are applied.

In most cases, these methods use meshes where the elements are fitted exactly to the material geometry in the sense that the element interfaces conform to the material boundaries or material interfaces. Such meshes are referred to as conforming meshes. However, the generation of conforming meshes is computationally expensive, especially if very complex-shaped objects are meshed. In the case of moving materials, where either a mesh adaption technique or a complete remeshing has to be performed at each point in time, the use of conforming meshes is even more computationally expensive.

In this work we propose a novel numerical method for field problems with either non-moving or moving complex-shaped materials. To avoid the computationally costly steps of generating conforming meshes and adapting these conforming meshes to moving materials, the proposed numerical method uses a structured fixed Cartesian mesh. To obtain accurate simulation results on such a static Cartesian mesh, an additional approach that accurately treats the material boundaries or interfaces is applied to the elements at the material boundaries or interfaces.

Our numerical method is embedded in a standard DG framework, since this framework allows for high order approximations and offers a huge flexibility when, as in our case, additional approaches need to be integrated into the standard framework. The numerical method is inspired by other numerical methods such as the Partially Filled Cell (PFC) approach for FIT introduced in [80], the locally conformal boundary FDTD algorithm in [23] or the fictitious domain methods for the Nitsche's method and the DG method presented in [16], [48] and [29]. These methods also accurately approximate field problems with complex-shaped materials on simple grids by using additional approaches that yield accurate approximations at the material boundaries or interfaces. A more detailed literature review is presented in Chapter 3.

The method presented in this work can potentially be applied to most electromagnetic (EM) field problems. However, in this work we restricted ourselves to electro-quasistatic (EQS) field problems and electrostatic (ES) field problems alike where either non-moving or moving complex-shaped materials are present. We first present the method in the context of EQS or ES field problems with non-moving materials. We label this approach as boundary conformal DG (BCDG) method. In a second step, we extend the BCDG method to electro-quasistatic field problems with moving materials. This extension of the BCDG method is called EBCDG method. The basic idea of our approach is illustrated in Figure 1.1 showing a water droplet oscillating on the insulation layer of a high voltage insulator. The computational domain consists of the water droplet, the insulation layer and air surrounding the droplet. The black lines represent an exemplary Cartesian mesh that covers the computational domain. As illustrated in Figure 1.1, the water droplet surface subdivides several elements into smaller sub-elements. These sub-elements are denoted as cut-cells in the following and are considered as independent elements in the mesh. If we assume a non-moving water droplet, the computational mesh with the additional cut-cells will remain constant during the simulation. However, if the water droplet oscillates over time, the water droplet takes different shapes at each point in time. Such a deformation of the water droplet is illustrated in Figure 1.1 where the water droplet is shown at two different points in time. Since the oscillation of the water droplet only affects a small number of cut-cells, it is not necessary to generate a whole new mesh. Instead, the discretization is modified locally by recalculating only the new cut-cells at each point in time while the underlying Cartesian grid is kept fixed. Hence, this method combines the accuracy of high-order approximations with the simple implementation and numerical efficiency of Cartesian grids.

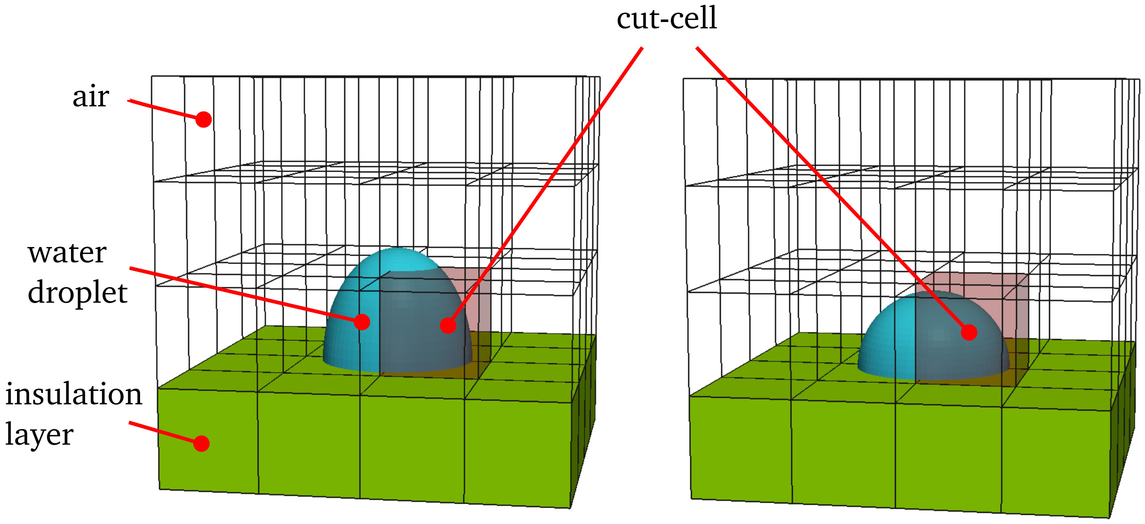

Figure 1.1.: Cartesian grid and droplet model at two different time levels. The grey shaded cells represent cut-cells intersected by the droplet boundary.

1.2 Contributions

The main contribution of this thesis is the development of the BCDG method and its extension, the EBCDG method. Both methods are based on a cut-cell approach to account for complex-shaped material boundaries on structured Cartesian meshes. As the numerical examples presented in this work show, both methods enable a accurate simulation of field problems with complex-shaped materials that may even move over time. Similar ideas for the approximation of such field problems have already been presented in several papers. Among the best-known methods in this respect are the Partially Filled Cell (PFC) approach for FIT in [80], the locally conformal FDTD algorithm in [23] or the Finite Volume approach using cut-cells in [66]. Furthermore, for the DG method and the Nitsche's method several approaches exist which operate on a simple mesh and use additional approaches at the vicinity of material interfaces or boundaries (see, e.g. [48], [29], [9], [57], [45]).

The BCDG method presented in this work is similar to the just mentioned methods in the sense that we also consider a discretization where elements at the material boundaries or interfaces are split into sub-elements. However, our method differs in many ways with respect to other methods: First the BCDG method is applied to the electro-quasistatic and the electrostatic approximation of the Maxwell's equations in a three-dimensionsional setting. Second, a DG method with the Local Discontinuous Galerkin (LDG) flux formulation is used. Third, specific hierarchical higher order approximation functions up to order three are used. Fourth, the BCDG method employs a different geometry representation and a different numerical integration compared to other methods. Moreover, several supplementary methods to the BCDG method and the EBCDG method are introduced in this work. The *adaptive approximation order method* and the *cell merging method* enable to deal with geometrically small cut-cells, whereas the DG hybridization has been developed to lower the number of degrees of freedom (DOF). All three methods can be applied on top of the cut-cell discretization approach. Up to our knowledge they are, with one exception, new in the literature on electromagnetic field problems with complex-shaped materials. Only very recently, an approach similar to the *cell merging method* has been proposed in [48]. However, at the time when [48] was published, our *cell merging method* had already been developed and a paper describing this method had been accepted for publication (see, e.g. [32]).

Finally, the extension of the BCDG method, the EBCDG method, represents a novel contribution to the literature on electromagnetic field problems with moving materials. Up to our knowledge, it is the first contribution in the field of electromagnetic which presents a cut-cell approach for moving material distributions. The reader is referred to Chapter 3 for a more detailed literature review. Furthermore, the reader is referred to [30], [32] and [31] were parts of this work were published.

1.3 Outline

This section presents a short outline of this work. In Chapter 2, we introduce the theory of electromagnetic fields. We show the full system of Maxwell's equations and the constitutive laws which completely describe the physical relations and interactions of electromagnetic fields. Since the focus in this work is on the ES and EQS approximations of the Maxwell's equations, we introduce both approximations in greater detail and describe the respective conditions which have to be satisfied to justify the application of these approximations. Furthermore, we show that the field problem of a water droplet oscillating on a high voltage insulator satisfies the classification conditions for EQS field problems. Finally, the definition of the energy and power dissipation for both the EQS and ES approximation of the Maxwell's equations are introduced.

In Chapter 3, the numerical framework of the BCDG and the extension of the BCDG (EBCDG) method is presented in greater detail. Since the BCDG and the EBCDG method are embedded in a DG method for standard grids, we begin by introducing the DG method in Section 3.2. We

derive the weak DG formulation for EQS and ES field problems and discuss the properties as well as the advantages and disadvantages of the DG method.
In Section 3.3, we present the details of the BCDG approach. The BCDG method comprises several supplementary approaches which are successively discussed. We start with the *cut-cell discretization approach*, followed by the *adaptive approximation order method*, the *cell merging method* and the *DG hybridization*. Since the BCDG method is only applied to the spatial derivatives of the EQS approximation of the Maxwell's equation, the weak DG formulation still contains the time derivative. The numerical methods that are used to discretize the time derivative of the weak DG formulation are discussed in Section 3.4. In Section 3.6, the extension of the BCDG (EBCDG) method to field problems consisting of materials that move over time is introduced. We show in detail how the changing material geometries are treated within the EBCDG framework. In Chapter 4, various numerical simulation results of the BCDG method and the EBCDG method are presented. In Section 4.1, the simulation results of the BCDG method on a verification example of a cylindrical capacitor filled with two dielectric material layers are shown. Several convergence studies and a comparison study with simulation results obtained from the commercial software CST STUDIO SUITE®2012 [1] demonstrate the high accuracy of the BCDG method when applied to the verification example. Moreover, the influence of the *adaptive approximation order method*, the *cell merging method* and the *DG hybridization* on the numerical results of the verification example are discussed. In Section 4.2, the numerical results of the EBCDG method are presented. The EBCDG method is applied to the example of an oscillating water droplet. In Section 4.2.2, we show the potential and electric field distribution of the EBCDG method for this example. In Section 4.2.3, we present a convergence study of the EBCDG method with respect to the energy and power dissipation obtained with the EBCDG method on the example. Chapter 5 summarizes the main features of the BCDG and the EBCDG method and briefly comments on possible extensions.

2 Electromagnetics

This chapter gives an overview on the theory of electromagnetic fields. We first introduce the Maxwell's equations and the associated constitutive laws which completely describe the physical relation and interactions of the electric and magnetic fields. Then, two approximations of the Maxwell's equations are presented, namely the electrostatic (ES) and the electro-quasistatic (EQS) approximation. We introduce both approximations because the numerical method developed in this work is applied to ES as well as EQS field problems. In addition, the conditions which classify a field problem as being ES or EQS are discussed. Finally, the energy and dissipation of ES and EQS approximations are specified.

2.1 Maxwell's equations and Constitutive equations

The famous Maxwell's equations built the foundation of the theory of electromagnetics. They completely describe electromagnetic interactions and are named after the famous Scottish physicist and mathematician James Clerk Maxwell (1831-1879)[82]. The equations partly result from the contributions of many famous scientists in the 18th and 19th century. However, it was James Clerk Maxwell who first published the full form of the Maxwell's equations in [58] in 1873. The modern form of the Maxwell's equations goes back to Oliver Heaviside who simplified the original set of 20 equations of James Clerk Maxwell to four equations in 1884 by using the mathematical discipline of vector calculus (see, e.g. [82] and [61]).
The Maxwell's equations in differential form are given by [44]:

$$\begin{aligned}
\nabla \times \mathbf{E}(\mathbf{x},t) &= -\frac{\partial \mathbf{B}}{\partial t}(\mathbf{x},t), && (2.1)\\
\nabla \times \mathbf{H}(\mathbf{x},t) &= \frac{\partial \mathbf{D}(\mathbf{x},t)}{\partial t} + \mathbf{J}(\mathbf{x},t), && (2.2)\\
\nabla \cdot \mathbf{D}(\mathbf{x},t) &= \rho(\mathbf{x},t), && (2.3)\\
\nabla \cdot \mathbf{B}(\mathbf{x},t) &= 0, && (2.4)
\end{aligned}$$

where the vector $\mathbf{x} \in \mathbb{R}^3$ represents the spatial parameter and the scalar $t \in \mathbb{R}$ represents the time dependency. Furthermore, the electromagnetic quantities with the associated SI units are denoted as:

$\mathbf{E}(\mathbf{x},t)$	*electric field strength*	SI unit: Vm^{-1}
$\mathbf{H}(\mathbf{x},t)$	*magnetic field strength*	SI unit: Am^{-1}
$\mathbf{D}(\mathbf{x},t)$	*electric displacement flux density*	SI unit: $Cm^{-2} = Asm^{-2}$
$\mathbf{B}(\mathbf{x},t)$	*magnetic flux density*	SI unit: $T = Vsm^{-2}$
$\mathbf{J}(\mathbf{x},t)$	*current density*	SI unit: $Cm^{-2}s^{-1} = Am^{-2}$
$\rho(\mathbf{x},t)$	*charge density*	SI unit: $Cm^{-3} = Asm^{-3}$

For the sake of simplicity, the spatial parameter $\mathbf{x} \in \mathbb{R}^3$ and the time parameter $t \in \mathbb{R}$ are generally omitted in the following chapters. Whenever the parameters are required, $\mathbf{x}$ and t are explicitly stated.

Equation (2.1), (2.2), (2.3) and (2.4) are called *Faraday's law*, *Ampere's law*, *Gauss's law* and *Gauss's law of magnetic flux density*, respectively. *Faraday's law* and *Ampere's law* of the Maxwell's equation indicate that the electric and the magnetic field are coupled. This is because *Faraday's law* includes the magnetic induction term and *Ampere's law* includes the displacement current term. Therefore, the curl of the electric field in *Faraday's law* influences the magnetic field and, vice versa, the curl of the magnetic field in *Ampere's law* affects the electric field through the magnetic induction and the displacement current terms [37].

The differential form of the Maxwell's equation can be transformed to the integral form by integrating (2.1) and (2.2) over a particular surface A in $\mathbb{R}^3$ and (2.3) and (2.4) over a particular volume V in $\mathbb{R}^3$. The *integral theorem of Stokes*, given by

$$\int_A \nabla \times \mathbf{E} \cdot d\mathbf{A} = \int_{\partial A} \mathbf{E} \cdot d\mathbf{s}, \tag{2.5}$$

and the *integral theorem of Gauss*, given by

$$\int_V \nabla \cdot \mathbf{D}\, dV = \int_{\partial V} \mathbf{D} \cdot d\mathbf{A}, \tag{2.6}$$

are then applied to (2.1)-(2.4) (see, e.g. [37]). The Maxwell's equations in integral form read [90]:

$$\int_{\partial A} \mathbf{E} \cdot d\mathbf{s} = -\int_A \frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{A}, \tag{2.7}$$

$$\int_{\partial A} \mathbf{H} \cdot d\mathbf{s} = \int_A \left(\frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}(\mathbf{x}, t) \right) \cdot d\mathbf{A}, \tag{2.8}$$

$$\int_{\partial V} \mathbf{D} \cdot d\mathbf{A} = \int_V \varrho\, dV, \tag{2.9}$$

$$\int_{\partial V} \mathbf{B} \cdot d\mathbf{A} = 0. \tag{2.10}$$

Note that the differential form and the integral form of the Maxwell's equations are equivalent. As mentioned before, both systems of equations (2.1)-(2.4) and (2.7)-(2.10) completely describe all known electromagnetic field phenomena (see, e.g. [37]). The main differences between the two forms emerge when they are applied to real-world examples (see, e.g. [37]). For examples with materials of complex shape, the application of the differential form of the Maxwell's equations is often easier, because the differential form applies to each point in space $\mathbf{x} \in \mathbb{R}$, whereas the integral form applies to certain volumes, surfaces and contours [37].

In this work, we focus on field problems with complex-shaped materials which cannot be solved analytically. Therefore, we apply a particular grid-based numerical method which is solely developed for the differential form of the Maxwell's equations.

Next, we would like to introduce another relation between the current and the charge density which is used several times in this work. This relation is denoted as the *charge conservation* and is obtained by entering *Gauss's law* into the divergence of *Ampere's law*. The charge conservation in integral and differential form[1] is given by

$$\oint_{\partial V} \mathbf{J} \cdot d\mathbf{A} + \frac{d}{dt} \int_{\partial V} \rho dV, \qquad \nabla \cdot \mathbf{J} + \frac{\partial \rho}{\partial t} = 0, \tag{2.11}$$

respectively. Next, the *constitutive equations* are introduced which specify the relation between the just introduced field quantities **D**, **E**, **B** and **H**. The *constitutive equations* are defined by

$$\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}, \tag{2.12}$$

$$\mathbf{B} = \mu_0 \mathbf{H} + \mathbf{M}, \tag{2.13}$$

where the material constants $\epsilon_0 = 8.854187 \cdot 10^{-12} As/Vm$ and $\mu_0 = 1.256 \cdot 10^{-6} Vs/Am$ represent the *permittivity* and *permeability of vacuum*, respectively [53]. The vectors **P** and **M** denote the *polarization density* and the *magnetization density* of the given materials, respectively. The *permittivity* and *permeability of vacuum* satisfy the following relation:

$$c = 1/\sqrt{\epsilon_0 \mu_0}, \tag{2.14}$$

where $c = 2.997924 \cdot 10^8 ms^{-1}$ is the speed of light. The *polarization* and the *magnetization densities*, **P** and **M**, are in general frequency, time, space and field dependent vector fields which illustrate the macroscopic behaviour of the material [53].

The materials used in this work are isotropic, paramagnetic or diamagnetic materials which can be considered as electrically and magnetically linear [37]. Furthermore, we assume a relaxation time which is much smaller than the period of interest. In this specific case, the *polarization density* **P** and the *magnetization density* **M** are linear functions of **E** and **H**, respectively. The *polarization density* **P** is defined as $\mathbf{P} = \epsilon_0 \chi_e \mathbf{E}$, where χ_e is the *dielectric susceptibility*. A similar relation holds for the *magnetization density*. Similarly, the *magnetization density* is given by $\mathbf{M} = \mu_0 \chi_m \mathbf{H}$, where χ_m represents the *magnetic susceptibility* [53]. Since we restrict ourselves to electrically and magnetically linear material, the constitutive equations (2.12)-(2.13) can be written as

$$\mathbf{D} = \epsilon \mathbf{E} = \epsilon_0 (1 + \chi_e) \mathbf{E} = \epsilon_0 \epsilon_r \mathbf{E}, \tag{2.15}$$

$$\mathbf{B} = \mu \mathbf{H} = \mu_0 (1 + \chi_m) \mathbf{H} = \mu_0 \mu_r \mathbf{H}, \tag{2.16}$$

where ϵ and μ represent the *permittivity* and *permeability* of the material, respectively. The *relative permittivity* and *relative permeability* of the material are given by $\epsilon_r = (1 + \chi_e)$ and $\mu_r = (1 + \chi_m)$, respectively.

Furthermore, we introduce the *conduction constitutive equation* which is also known as *Ohm's law*. *Ohm's law* establishes the link between the *conduction current density* $\mathbf{J}_l$ and the electric field **E** and is defined by

$$\mathbf{J}_l = \kappa \mathbf{E}, \tag{2.17}$$

where κ denotes the *electric conductivity* of the material (SI unit: Siemens/m=S/m) . The *current density* is given by

$$\mathbf{J} = \mathbf{J}_l + \mathbf{J}_s + \mathbf{J}_c, \tag{2.18}$$

[1] The differential form of the *charge conservation* is also referred to as *continuity equation*.

where $\mathbf{J}_s$ and $\mathbf{J}_c$ are the *source current density* and the *convection current density*, respectively. The *conduction current density* is induced by the electric field in materials which have electric conductivities as shown in (2.17). The *convection current density* $\mathbf{J}_c = \rho\mathbf{v}$ is caused by free charges which have the charge density ρ and move with a velocity $\mathbf{v}$ in a neutral background such as vacuum [90]. The source current density is field independent and induced by current sources [37].

Finally, we introduce the continuity conditions of the field quantities **E**, **H**, **D** and **B** at material interfaces. If the field problem contains certain piecewise uniform materials, e.g. material a and b, the continuity conditions at an exemplary material interface of a and b in $\mathbb{R}^3$ read:

$$\mathbf{n} \times (\mathbf{E}_a - \mathbf{E}_b) = \mathbf{0}, \tag{2.19}$$

$$\mathbf{n} \cdot (\mathbf{D}_a - \mathbf{D}_b) = \sigma, \tag{2.20}$$

$$\mathbf{n} \times (\mathbf{H}_a - \mathbf{H}_b) = \mathbf{J}_F, \tag{2.21}$$

$$\mathbf{n} \cdot (\mathbf{B}_a - \mathbf{B}_b) = 0. \tag{2.22}$$

The vector **n** represents the interface normal, $\mathbf{J}_F$ a *surface current density* and σ a *surface charge density* at the material interface. These conditions can be derived by applying the Maxwell's equations in (2.7)-(2.10) to a small domain very close to an interface of the materials a and b [90].

2.2 Classification of Electromagnetic Field Problems

For many field problems, approximations of the full system of Maxwell's equations in (2.1)-(2.4) are sufficient to accurately describe particular field problems. Several different approximations of the full system of Maxwell's equations (2.1)-(2.4) have been developed, namely the *electrostatic*, *electro-quasistatic*, *magnetostatic*, *magneto-quasistatic*, *hybrid-quasistatic* and *stationary current field approximation* [50]. It can be shown that all of these approximations are justified for certain application examples. In this work, we are concerned with examples which can be sufficiently described using the *electro-quasistatic* (EQS) or the *electrostatic* (ES) approximation of the Maxwell's equations. Hence, only these two approximations of the Maxwell's equations are presented below. The reader is referred to literature for further information about the other field approximations (see, e.g. [44], [90], [37], [50], [70] and [85]).

2.2.1 Electrostatic approximation of the Maxwell's equations

The *electrostatic* (ES) approximation of the Maxwell's equations is applied to real-word applications which are mainly characterized by stationary charge distributions [90]. The field quantities of these application examples are considered as time invariant, implying that the time derivatives of the Maxwell's equations can be omitted. Furthermore, the current densities in ES field application are assumed to be zero, i.e. $J_c = 0$, $J_s = 0$ and $J_l = \kappa\mathbf{E} = 0$ [90].

In this case, the Maxwell's equations decouple, i.e. the electric and magnetic field quantities do not interact. As a result, the Maxwell's equations split in two parts, the ES approximation and the *magnetostatic* (MS) approximation. Therefore, they can be considered independently of each other.

The ES approximation of the Maxwell's equations reads

$$\nabla \times \mathbf{E} = \mathbf{0}, \tag{2.23}$$

$$\nabla \cdot \mathbf{D} = \rho. \tag{2.24}$$

Since the right-hand side of (2.23) is set to zero, the approximation is regarded as irrotational[2] and a scalar function can be introduced which simplifies the ES formulation [37]. In this case, a scalar potential function $\phi(\mathbf{x})$ can be specified that satisfies

$$\phi(\mathbf{x}) - \phi(\mathbf{x}_{ref}) = \int_{\mathbf{x}}^{\mathbf{x}_{ref}} -\mathbf{E} \cdot d\mathbf{s}. \tag{2.25}$$

To uniquely determine the potential function $\phi(\mathbf{x})$ - often referred to as potential of $\mathbf{x}$ - for all points $\mathbf{x} \in \mathbb{R}^3$, a reference point needs to be specified. In general, this specification depends on the particular field problem. However, in many cases the reference point is set to infinity.
Note that the potential ϕ satisfies

$$\mathbf{E} = -\nabla \phi. \tag{2.26}$$

Entering (2.26) in (2.24) simplifies the ES approximation in (2.23)-(2.24). The ES approximation is reduced to a single equation which is given by

$$\nabla \cdot \epsilon \nabla \phi = -\rho. \tag{2.27}$$

Equation (2.27) belongs to the class of linear second order partial differential equations (PDE). It can be shown that a solution of (2.27) exists and is uniquely defined if the boundary conditions are well-defined [37]. For this reason, equation (2.27) can also be considered as a boundary value problem.
If field problems containing a single isotropic linear material are considered, the ES approximation can be written in the form

$$\nabla \cdot \nabla \phi = \Delta \phi = -\rho / \epsilon. \tag{2.28}$$

Equation (2.28) is also referred to as *Poisson's equation*. In the special case of $\rho = 0$, (2.28) is denoted as *Laplace's equation*.

2.2.2 Electro-quasistatic approximations of the Maxwell's equations

As mentioned before, the full system of Maxwell's equations completely describe all possible relations and interactions of electric and magnetic fields [82]. This also holds true for the most complex wave phenomena at high frequencies and on short time scales [37]. However, if electromagnetic waves propagate through the field domain in a period of time that is much shorter than the time of interest, the electromagnetic wave effects become irrelevant and can be neglected [37]. In this case, quasistatic approximations of the Maxwell's equations can be applied to describe the electromagnetic behaviour of the field problem. Several quasistatic

[2] Such an approximation is often also called *eddy-current free*.

approximations exist. The most well-known approximations are the *electro-quasistatic* (EQS) and the *magneto-quasistatic* (MQS) approximations. Furthermore, *hybrid-quasistatic* approximations^{34} exist, which consider both the capacitive and inductive effects of the EQS and MQS approximations, respectively [50].

In this work, we are only concerned with the EQS approximation of the Maxwell's equations. Therefore, only the EQS approximation is presented in the following. Furthermore, we explain why the EQS approximation is more suitable in some cases than the full system of Maxwell's equation and introduce the conditions field problems have to satisfy to justify the EQS approximation. Further information on the MQS and the *hybrid-quasistatic* approximation can be found in [37], [24], [50] and [70].

The EQS approximation is obtained by neglecting the magnetic induction term $\partial \mathbf{B}/\partial t = 0$ in *Faraday's law* (2.1). Hence, the EQS approximation of the Maxwell's equations in the differential form is given by:

$$\begin{aligned} \nabla \times \mathbf{E}(\mathbf{x}, t) &= 0, && (2.29) \\ \nabla \times \mathbf{H}(\mathbf{x}, t) &= \frac{\partial \mathbf{D}(\mathbf{x}, t)}{\partial t} + \mathbf{J}(\mathbf{x}, t), && (2.30) \\ \nabla \cdot \mathbf{D}(\mathbf{x}, t) &= \varrho(\mathbf{x}, t), && (2.31) \\ \nabla \cdot \mathbf{B}(\mathbf{x}, t) &= 0. && (2.32) \end{aligned}$$

By setting the term $\partial/\partial t\mathbf{B}$ to zero, the electromagnetic wave effects of the Maxwell's equations are neglected. As a result, *Faraday's law* implies that the electric field becomes irrotational. Furthermore, equation (2.29) and (2.31) uniquely determine the electric field, if the charge density is given [37]. However, this is not the case in general. If the charge density is not known, it can be eliminated by applying the divergence to equation (2.31) which yields

$$\nabla \cdot \frac{\partial}{\partial t}\mathbf{D} + \nabla \cdot \mathbf{J} = 0. \tag{2.33}$$

In this case, (2.29) and (2.33) completely specify the electric field [37]. Having determined the electric field, the magnetic field can be obtained from (2.30) and (2.32).

Note that EQS fields are sometimes described as fields that proceed from one point in time to the next as if they are static [37]. This behaviour might result from the fact that the source distribution in (2.29) and (2.31) at each point in time uniquely determines the whole electric field distribution at the same point in time [37].

As in the ES case, the electric field is irrotational, which implies that a scalar potential function $\phi(\mathbf{x})$ can be used to simplify the formulation. Hence, equation (2.29) can be written in the form

$$\mathbf{E} = -\nabla\phi. \tag{2.34}$$

As shown in Section 2.2.1, to uniquely determine the potential function an additional reference point has to be specified [37].

[3] In [70], the *hybrid-quasistatic* approximation is referred to as *Lorenz-Quasi-Static Formulation*. It is also known as *Quasi-Static* or *Electro-Magneto-Quasi-Static* regime (see, e.g.[4]) and [94]).

[4] For interacting charged particles in free space, these approximations are also known as *Darwin formulation* [70].

In the following, we refer to (2.33)-(2.34) as mixed EQS formulation of the Maxwell's equations. Nonetheless, using equation (2.34) in (2.33), the mixed EQS mixed formulation can be reduced to a single equation:

$$\nabla \cdot \left(\kappa \nabla \phi + \frac{\partial}{\partial t} \epsilon \nabla \phi \right) = 0. \tag{2.35}$$

For any given well-defined combination of initial and boundary conditions, equation (2.35) uniquely determines the electric field [37]. Since in addition initial conditions are necessary to uniquely determine the solution, the EQS approximation is regarded as a *initial and boundary value problem*. Finally, note that equation (2.35) belongs to the class of linear parabolic second order PDE [84].

2.2.2.1 Classification of electro-quasistatic field conditions

Slowly oscillating field problems raise the question whether to use an EQS approximation, a MQS approximation, a hybrid-quasistatic approximation or the full system of Maxwell's equations to describe the field problem. In this work we mainly focus on the EQS approximation of the Maxwell's equations. Therefore, we only introduce the conditions which have to be satisfied to justify the EQS approximation of the Maxwell's equations. However, before we derive the conditions for the EQS approximations, we first introduce two well-known examples of characteristic EQS and MQS field problems. In the first example, a pair of metal spheres which are insulated from each other and excited by a voltage source are considered. This example is often described as a typical EQS example [37]. The second, exemplary quasistatic example represents a perfectly conducting metal loop which is driven by a current source. This example represents a typical MQS field problem. However, most field problems are more complex than the just mentioned characteristic EQS and MQS field problems. Nevertheless, field problems which are classified as EQS and MQS generally appear to be similar to the just mentioned EQS or MQS examples [37]. When dealing with a field problem which is very similar to the presented characteristic field problems, it often helps to just lower the frequency of the driving source and check the fields to determine whether it is EQS or MQS [37]. This procedure is sometimes referred to as a *rule of thumb* [37]. If the frequency is lowered to almost zero and the magnetic field vanishes, the field problem is very likely to be EQS. By contrast, if the electric field vanishes at a very low frequency, the problem is very likely to be MQS.

However, to thoroughly classify field problems as being EQS or MQS, more detailed classification conditions need to be considered. To derive these classification conditions, we have to consider the error fields of the EQS approximation. In general, an EQS approximation is justified, if the error caused by neglecting the magnetic induction term $\partial / \partial t \mathbf{B}$ is small compared to the fields of the field problem. Note that this error depends on the problem size length l of the field problem. In this work, the length l of a field problem domain is defined by $l = \max l_i$, where l_i are the size lengths in each axis direction. Furthermore, the error fields also depend on several characteristic times which we introduce next.

First, we present the *electro-quasistatic charge relaxation time* τ_e. It describes the relaxation behaviour of free charges and is defined by (see, e.g. [37])

$$\tau_e = \frac{\epsilon}{\kappa}. \tag{2.36}$$

Second, the *magneto-quasistatic diffusion time* τ_m describes the time dependence of the current density **J** and the magnetic field **H** and is given as

$$\tau_m = \mu\kappa l^2. \tag{2.37}$$

In contrast to the charge relaxation time τ_e, the magnetic diffusion time τ_m depends on the size length l of the system. Further details can be found in [37].
The *electromagnetic wave transit time* τ_{em} is defined by

$$\tau_{em} = \frac{l}{c} = l\sqrt{\mu\epsilon} = \sqrt{\tau_e \tau_m}. \tag{2.38}$$

This characteristic time represents the time which an electromagnetic wave requires to propagate at velocity $c = 1/\sqrt{\epsilon\mu}$ through a material domain with problem size length l [37] (see Section 2.1). Note that the definition of the *electromagnetic wave transit time* implies that the *electromagnetic wave transit time* τ_{em} always lies between the two characteristic times τ_e and τ_m [37]. For that reason, either $\tau_e < \tau_{em} < \tau_m$ or $\tau_m < \tau_{em} < \tau_e$ is satisfied.
Finally, we introduce the *characteristic time of the excitation* τ which is the driving source of the system. In this work we assume a sinusoidal excitation. The characteristic time of the sinusoidal excitation is given by

$$\tau = \frac{1}{\omega}, \tag{2.39}$$

where $\omega = 2\pi f$ is the angular frequency (SI unit: rad/s) and f the frequency (SI unit: Hz) of the sinusoidal excitation.
The characteristic times and the angular frequency ω can now be used to transform the Maxwell's equations to a normalized formulation. This normalized formulation requires that the spatial coordinates and the time are also transformed to a normalized form. The normalized spatial coordinates and the normalized time are defined as follows:

$$\mathbf{x} = (x,y,z)^T = (\bar{x}l, \bar{y}l, \bar{z}l)^T, \qquad t = \bar{t}/\omega, \tag{2.40}$$

where the bar denotes the normalized variables. Furthermore, the electric field **E**, the magnetic field **H** and the charge density ρ are normalized using the following expressions:

$$\mathbf{E} = E_{ref}\bar{\mathbf{E}}, \qquad \mathbf{H} = E_{ref}\sqrt{\frac{\epsilon}{\mu}}\bar{\mathbf{H}}, \qquad \rho = \frac{\epsilon E_{ref}}{l}\bar{\rho}, \tag{2.41}$$

where E_{ref} represents a typical field intensity [37]. Entering the normalized entities and the characteristic times into (2.36)- (2.39) yields the following normalized formulation of the full system of the Maxwell's equation [37]:

$$\bar{\nabla} \times \bar{\mathbf{E}} = -\omega\tau_{em}\frac{\partial \bar{\mathbf{H}}}{\partial \bar{t}}, \tag{2.42}$$

$$\bar{\nabla} \times \bar{\mathbf{H}} = \omega\tau_{em}\left(\frac{1}{\omega\tau_e}\bar{\mathbf{E}} + \frac{\partial \epsilon\bar{\mathbf{E}}}{\partial \bar{t}}\right), \tag{2.43}$$

$$\bar{\nabla} \cdot \epsilon\bar{\mathbf{E}} = \bar{\rho}, \tag{2.44}$$

$$\bar{\nabla} \cdot \mu\bar{\mathbf{H}} = 0. \tag{2.45}$$

Note that the parameters $\omega\tau_m$, $\omega\tau_e$ and $\omega\tau_{em}$ in (2.42)-(2.45) are dimensionless and determine the fields of the system [37]. Therefore, they are used to justify the EQS approximation of the Maxwell's equations. In the following we assume a field problem with specified uniform material parameters and size length l. This implies that the field problem has the three dimensionless parameters $\omega\tau_{em}$, $\omega\tau_e$ and $\omega\tau_{em}$.

To demonstrate how the dimensionless parameters determine the field distributions, we present the two possible orderings of the characteristic times. As mentioned previously, from the definition of the *electromagnetic wave transit time* τ_{em} it follows that we have either $\tau_e < \tau_{em} < \tau_m$ or $\tau_m < \tau_{em} < \tau_e$. Figure 2.1 illustrates these possible ordering of the reciprocal of the characteristic times with respect to the angular frequency of the applied excitation. The case $\tau_m < \tau_{em} < \tau_e$ is represented on the left-hand side of Figure 2.1, whereas the case $\tau_e < \tau_{em} < \tau_m$ is shown on the right-hand side of Figure 2.1. The red and blue shaded domain on the left-hand and right-hand side of Figure 2.1 illustrate the domains where a EQS and MQS approximation is justified [37]. Next, we explain how the dimensionless parameters $\omega\tau_{em}$, $\omega\tau_e$ and $\omega\tau_{em}$ are used to justify the EQS approximation. For this purpose, we assume that the ordering of the characteristic times of our field problem is equivalent to the ordering presented on left-hand side of Figure 2.1, i.e. $\tau_m < \tau_{em} < \tau_e$. If the dimensionless parameters $\omega\tau_e$ and $\omega\tau_{em}$ of our field problem satisfy $\omega\tau_e > 1$ and $\omega\tau_{em} < 1$, equation (2.43) implies that the term $\omega\tau_{em}\partial\epsilon\bar{\mathbf{E}}/\partial\bar{t}$ is the dominant term on the right-hand side of (2.43). This, in turn, implies that the magnetic field $\bar{\mathbf{H}}$ is approximately of order $\omega\tau_{em}\bar{\mathbf{E}}$. Entering this approximation in *Faraday's law* in (2.42) yields a magnetic induction term which is approximately of order $(\omega\tau_{em})^2\bar{\mathbf{E}}$. Hence, we conclude that whenever $(\omega\tau_{em})^2 << 1$, the magnetic induction term on the right-hand side term of (2.42) is very small compared to the field terms. Therefore, the error caused by neglecting the magnetic induction term on the right-hand side term of (2.42) is also very small compared to the field terms. Thus, the use of the EQS approximation is justified in this case.

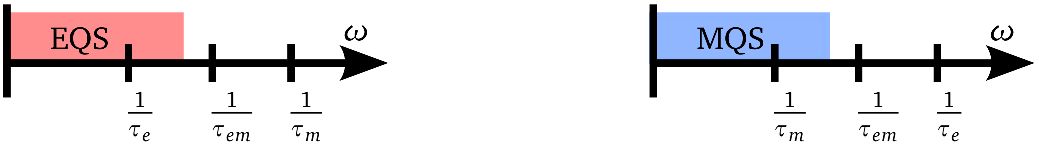

Figure 2.1.: Ordering of reciprocal of characteristic times with respect to the angular frequency. Source: [37]

Next, we consider the case where we have $\tau_m < \tau_{em} < \tau_e$ and $\omega\tau_e < 1$ and $\omega\tau_{em} < 1$. Again, equation (2.43) implies that the dominant term on the right-hand side of (2.43) is the term $\tau_{em}/\tau_e\bar{\mathbf{E}}$. Therefore, the magnetic field $\bar{\mathbf{H}}$ is of order $\tau_{em}/\tau_e\bar{\mathbf{E}}$ and the magnetic induction term on the right-hand side term of (2.42) is of order $(\omega\tau_{em}^2/\tau_e)\bar{\mathbf{E}}$. We conclude that the error caused by neglecting magnetic induction term on the right-hand side term of (2.42) is very small compared to the field terms, if $(\omega\tau_{em}^2/\tau_e) << 1$. In this case, the use of the EQS approximation is also justified.

The red shaded domain on the left-hand side of Figure 2.1 illustrates both above described cases: either we have $\omega\tau_e > 1$ and $\omega\tau_{em} < 1$ with $(\omega\tau_{em})^2 << 1$ or $\omega\tau_e < 1$ and $\omega\tau_{em} < 1$ with $(\omega\tau_{em}^2/\tau) << 1$. Figure 2.1 furthermore shows that if the angular frequency is increased, the EQS approximation is not longer justified. Increasing the angular frequency implies that electromagnetic wave effects come into play. The terms $(\omega\tau_{em})^2$, $(\omega\tau_{em}^2/\tau)$ and, therefore, the right-hand side term in (2.42) can no longer be considered as negligible. In this case, the field problem needs to be described by the full system of Maxwell's equations [37].

A similar argumentation holds for the MQS approximation which is illustrated on the right-hand side of Figure 2.1. The blue shaded domain illustrates the angular frequency domain which justifies the MQS approximation. Since we focus on the EQS approximation in this work, the reader is referred to the literature for more information on the MQS classification conditions (see, e.g. [37]).

So far, we have shown how the EQS approximation is justified for a fixed field problem size length l. In the following, we discuss the influence of the size l on the EQS and MQS justification. As shown in (2.37) and (2.38), the electromagnetic wave transit time τ_{em} and the magnetic diffusion time τ_m depend on the size length l. For that reason, the size length l also plays a role whether an EQS or MQS approximation is justified. The influence of the length l on the characteristic times is shown in Figure 2.2. In this figure the conditions of the dimensionless parameters, given by $\omega\tau_{em} = 1$, $\omega\tau_e = 1$ and $\omega\tau_m = 1$, are illustrated with respect to the size length (l/l^*) and the dimensionless parameter $\omega\tau_e$. In Figure 2.2 the parameter l^* denotes the *characteristic length* which is defined by

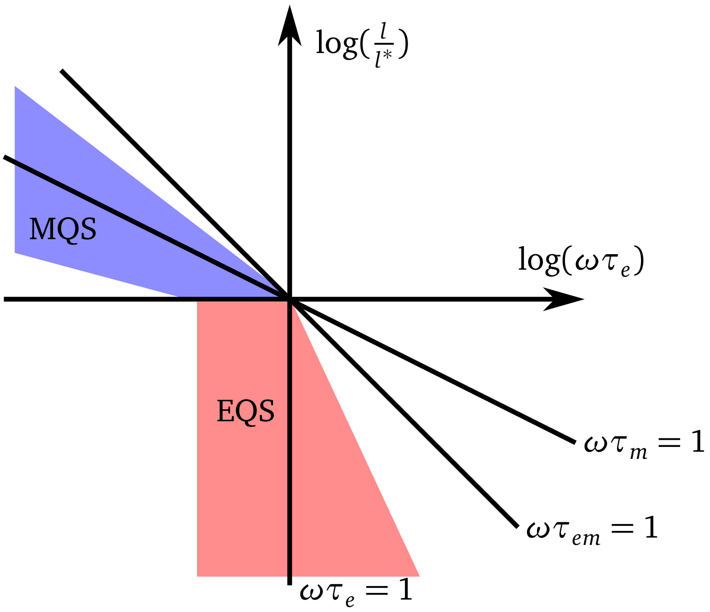

Figure 2.2.: Ordering of the dimensionless parameters $\omega\tau_{em} = 1$, $\omega\tau_e = 1$, $\omega\tau_m = 1$ with respect to $\omega\tau_e$ and the size length (l/l^*). Source: [37]

$$l^* = 1/\kappa\sqrt{\epsilon/\mu}. \tag{2.46}$$

Furthermore, the conditions of the dimensionless parameters, $\omega\tau_{em} = 1$ and $\omega\tau_m = 1$ are presented with respect to the length (l/l^*) and $\omega\tau_e$ using a normalization to $\omega\tau_e$. Therefore, $\omega\tau_{em} = 1$ and $\omega\tau_m = 1$ are written with respect to (l/l^*) and $\omega\tau_e$ as follows (see, e.g. [37]):

$$\omega\tau_e = 1, \tag{2.47}$$
$$\omega\tau_m = 1 \Rightarrow \omega\tau_e = (l/l^*)^{-2}, \tag{2.48}$$
$$\omega\tau_{em} \Rightarrow \omega\tau_e = (l/l^*)^{-1}. \tag{2.49}$$

Note that Figure 2.2 shows the parameters with respect to (l/l^*) and $\omega\tau_e$ using a log-log scale, because it allows to plot the conditions $\omega\tau_{em} = 1$, $\omega\tau_e = 1$, $\omega\tau_m = 1$ as straight lines. As mentioned previously, the values of the dimensionless parameters determine if a EQS and MQS approximation is justified. Recall that we have illustrated the two possible ordering of the reciprocal of the characteristic times in Figure 2.1. Both possible orderings presented in Figure 2.1 are also shown in Figure 2.2. For instance, the reciprocal ordering of the characteristic times with respect to the angular frequency ω shown on the right-hand side of Figure 2.1 is equivalent to the reciprocal ordering presented in Figure 2.2, if the parameter l satisfies $log(l/l^*) > 1$. By contrast, the ordering of the reciprocals of the characteristic times shown on the left-hand side of Figure 2.1 corresponds to the ordering illustrated in Figure 2.2, if $log(l/l^*) < 1$.

The red shaded and the blue shaded domains present the domains for which the EQS and MQS approximation is justified. These domains correspond to the EQS and MQS approximation domains presented in Figure 2.1. As mentioned previously, the EQS approximation is justified if the characteristic times satisfy $\tau_m < \tau_{em} < \tau_e$ with either $\omega\tau_e > 1$ and $\omega\tau_{em} < 1$ or $\omega\tau_e < 1$ and $\omega\tau_{em} < 1$ and if the magnetic induction term on the right-hand side of (2.42) is of order $(\omega\tau_{em})^2 << 1$ or $(\omega\tau_{em}^2)/\tau_e << 1$, respectively.

So far, we have restricted ourselves to field problems with uniform material parameters. However, most examples - including our water droplet example - consist of several materials domains. Each material domain is characterized by its own set of characteristic times. This implies that all characteristic times of the material domains have to be analysed. The EQS approximation is only justified, if the error caused by neglecting the magnetic induction term on the right-hand side term of (2.42) is small for all characteristic times of the material domains [37].

Another issue arises when dealing with field problems consisting of several material domains. Some field problems are dynamic in their behaviour, but can be split into subsystem which are similar to characteristic EQS or MQS field problems. This implies that dominant subsystems which mainly affect the electromagnetic behaviour have to be specified. Then, the classification can be performed as shown above using the characteristic times and the size length of the dominant subsystem.

Finally, we would like to emphasize that we could also check whether an EQS and MQS approximation is justified by evaluating the electric and magnetic energy densities of the field problem. This is due to the fact that the magnetic and electric energy densities take very low values or vanish in EQS and MQS systems, respectively [37]. The energy densities are introduced in Section 2.2.2.3. The reader is referred to Section 2.2.2.3 for more detailed information. However, in this work we only consider the classification conditions with respect to the characteristic times and the size length l.

In the next section, we will use the just introduced classification conditions to check whether the example of the water droplet on the high voltage insulators justifies the use of the EQS approximation of the Maxwell's equations.

2.2.2.2 Justification of the EQS approximation for the water droplet application example

In Chapter 1 we have claimed that the example of water droplets oscillating on the insulation layer of high voltage insulators can be described by the EQS approximation of the Maxwell's equations. In this section, we check whether the use of the EQS approximation of the Maxwell's equations is indeed justified for this example. For this purpose, we need to consider the characteristic times τ_e, τ_m, τ_{em} and τ and determine the size length l of the field problem. Note that we make use of experimental setup data obtained from experimental studies of the water droplet phenomena on high voltage insulators [49]. The experimental setup proposed in

[49] consists of a polymer insulation layer which contains two electrodes. A water droplet is placed on the polymer insulation layer and a voltage excitation of $1kV$ is applied to the electrodes at a working frequency of 50Hz. This implies that the angular frequency is given by $\omega = 2\pi \cdot 50 = 314.16$. The maximum size length of the setup is $l = 0.1m$.

In [49] distilled water droplets are placed on a polymer insulation layer which contains two electrodes. In Table 2.1 the material parameters, the characteristic size length l^* and the dimensionless parameters $\omega\tau_e$, $\omega\tau_{em}$ and $\omega\tau_m$ are listed for each material. Table 2.1 shows that the

materials	ϵ_r	μ_r	κ	l^*	$\omega\tau_e$	$\omega\tau_{em}$	$\omega\tau_m$
distilled water	80.18	1.0	6.4E-6	3.7E+3	3.5E-2	9,4E-7	2.5E-11
air	1.0	1.0	3.0E-15	8.9E+11	9.3E+5	1.1E-7	1.2E-20
polymer	2.8	1.0	1.0E-14	4.4E+11	7.8E+5	1.7E-7	4.0E-20

Table 2.1.: Material parameters, characteristic times and sizes of an experimental setup proposed in [49] for experimental studies of the water droplet phenomena on high voltage insulators

dimensionless parameters of the distilled water droplet, $\omega\tau_e$ and $\omega\tau_{em}$, are both smaller than one. Furthermore, the length $\log(l/l^*)$ is much smaller than one. This implies that the magnetic induction term on the right-hand side of (2.42) is approximately of order $(\omega\tau_{em}^2)/\tau_e\bar{\mathbf{E}}$. Since $(\omega\tau_{em}^2)/\tau_e = \omega\tau_m$ and since the parameter of the distilled water satisfies $\tau_m \approx$ 2.5E-11<< 1, the magnetic induction term $(\omega\tau_{em}^2)/\tau_e\bar{\mathbf{E}}$ is approximately of order 2.5E-11$\bar{\mathbf{E}}$. In contrast to distilled water, the dimensionless parameters of both the polymer insulation layer and the air satisfy $\omega\tau_e > 1$ and $\omega\tau_{em} < 1$. Therefore, the magnetic induction term on the right-hand side of (2.42) is approximately of order $(\omega\tau_{em})^2\bar{\mathbf{E}}$. For the polymer insulation layer we have $(\omega\tau_{em})^2 \approx$ 3.1E-14 and for the air we have $(\omega\tau_{em})^2 \approx$ 1.1E-14. Since the order of magnitude of the magnetic induction term is very small for all materials, the error caused by neglecting the magnetic induction term is negligible. To summarize, the necessary conditions for the application of the EQS approximation are satisfied for our water droplet example.

2.2.2.3 Energy and power dissipation density for the EQS and ES approximation of the Maxwell's equations

In this section, we present the energy conservation, energy density and power dissipation density of the EQS and the ES approximation of the Maxwell's equation. These quantities are introduced since they are used in a verification study of the BCDG method and the EBCDG method in Chapter 4. As we have mentioned in Section 2.1, we restrict ourselves in this work to isotropic, paramagnetic or diamagnetic materials which are considered as electrically and magnetically linear. Therefore, the energy conservation, energy and power dissipation density are introduced with respect to electrically and magnetically linear materials.

The *energy conservation* provides the link between the storage of energy and the energy which is transformed into other forms of energies such as heat or mechanical energy. The energy conservation in integral and differential form is defined by

$$-\oint_{\partial V} \mathbf{S} \cdot d\mathbf{A} = \frac{d}{dt}\int_V W dV + \int_V P_d dV, \qquad \nabla \cdot \mathbf{S} + \frac{\partial}{\partial t} W + P_d = 0, \tag{2.50}$$

where W represents the energy density (SI unit: $Joule/m^3 = J/m^3$), P_d the *power dissipation density* (SI unit: $Watt/m^3 = W/m^3$) and V an arbitrary volume in $\mathbb{R}^3$ [37]. The vector $\mathbf{S}$ is denoted as *Poynting vector* and represents the *energy flux density* (SI unit: W/m^2)(see, e.g. [85] or [37]). The *energy conservation* can be derived from the Maxwell's equations if (2.1) and (2.4) are dot multiplied by **E** and **H**, respectively. Adding up the two resulting equations yields equation (2.50). Generally speaking, equation (2.50) implies that the power that flows with *energy flux density* S into a volume is equal to the rate of increase of the total *energy* and the *power dissipation density* [37].

The *Poynting vector* S is defined by (see, e.g. [53])

$$\mathbf{S} = \mathbf{E} \times \mathbf{H}. \tag{2.51}$$

Furthermore, for the EQS approximation the *Poynting vector* S can also be expressed as (see, e.g. [37])

$$\mathbf{S} = \phi \left(\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \right). \tag{2.52}$$

The *energy density* reads

$$W = w_e + w_m, \tag{2.53}$$

where $w_e = 1/2\, \mathbf{E} \cdot \mathbf{D}$ is the *electric energy density* and $w_m = 1/2\, \mathbf{B} \cdot \mathbf{H}$ the *magnetic energy density* for isotropic, electrically and magnetically linear materials. Furthermore, the *power dissipation density* is given by

$$P_d = \kappa E \cdot E. \tag{2.54}$$

For EQS field problems the *electric energy density* w_e can be considered as the dominant density. Moreover, the *magnetic energy density* w_m is negligible in many cases. Note that for field problems having uniform material parameters and size length l the *magnetic* and *electric energy density* satisfy the following equation:

$$\frac{w_m}{w_e} = K \left(\frac{l}{l^*} \right)^2, \tag{2.55}$$

where l^* is the *characteristic length* introduced in Section 2.2.2.1 and K is a parameter which is of the order of unity [37]. As we have shown in Section 2.2.2.1, the EQS approximation is only justified for $\left(\frac{l}{l^*}\right) < 1$. This implies that the *magnetic energy density* w_m of EQS field problems is smaller than the *electric field density* w_e. Furthermore, the *magnetic energy density* is negligible if $\left(\frac{l}{l^*}\right) << 1$.

For the ES field problems, the *power dissipation density* term vanishes, i.e. $P_d = 0$, since the current density **J** is assumed to be equal to zero (see Section 2.2.1). Furthermore, since the potential and electric field distributions are assumed to be static, the energy conservation is obviously not applicable. However, for ES field problems and isotropic, electrically and magnetically linear materials the *energy density* is defined by

$$W = \frac{1}{2} \mathbf{E} \cdot \mathbf{D} \tag{2.56}$$

and the *total energy* of ES field problem is given as

$$W_t = \frac{1}{2} \int_V \mathbf{E} \cdot \mathbf{D} dV, \tag{2.57}$$

where V represents the ES field domain [90]. In Chapter 4 we use the *energy* and *power dissipation densities* to assess the accuracy of the BCDG method with respect to the energy. Furthermore, the convergence study of the EBCDG method is presented with respect to the *energy* and *power dissipation densities*.

3 Numerical Methods for Electro-quasistatic Field Problems

This chapter presents the numerical framework of this work. The first section of this chapter gives a brief overview of known numerical methods which are related to the numerical method developed in this thesis. Then, Section 3.2 introduces the DG method in greater detail. We focus on the Local DG (LDG) method which was first proposed in [20]. This DG method is applied only to the spatial derivatives of the EQS approximation of the Maxwell's equations. We will present the variational formulation, the semi-discrete weak formulation, and the numerical flux formulation of the DG method. Furthermore, important numerical properties and features such as stability, consistency and convergence of the DG method are discussed. In Section 3.3 the boundary conformal DG (BCDG) method is presented. The BCDG method is based on the standard DG framework and enables an accurate approximation of complex-shaped materials on structured Cartesian grids. Since the DG method is only applied to the spatial derivatives, the remaining time derivative in the semi-discrete weak DG formulation has to be discretized separately[1]. In Section 3.4 the numerical methods that are applied to the time derivative of the resulting semi-discrete electro-quasistatic Maxwell's equations are shown. The last section of this chapter introduces an extension of the boundary conformal DG (EBCDG) method to field problems including materials that move over time.

3.1 Introduction and literature overview

Over the last decades, many methods have been introduced to solve the Maxwell's equations and their approximations such as the electro-quasistatic Maxwell's equations. The most established numerical methods are the so-called grid-based discretization methods which operate on a partition of the computational domain into simpler elements[2]. The best-known grid-based numerical discretization methods to solve the Maxwell's equations are the *Finite Difference Time Domain* (FDTD) methods, the *Finite Integration Technique* (FIT), the *Finite Element* (FE) method and the *Discontinuous Galerkin* (DG) method.

The *Finite Difference Time Domain* (FDTD) method has first been published in [92] in 1966. As the name indicates, the method is based on the finite difference (FD) scheme for the Maxwell's equations in time domain where the electric and the magnetic field component are solved using a "leap-frog" scheme [85].

The *Finite Integration Technique* (FIT) was first introduced in 1977 in [91] and is a general method to solve the full set of Maxwell's equation in time and frequency domain. It can be applied to all kind of electrodynamic problems as well as to other application problems, for instance, in the field of accoustics or elastodynamics (see, e.g [89], [85]).

[1] This separation of the spatial discretization and time discretization is referred to as *method of lines* [86].

[2] For example, hexahedral or tetrahedral elements may be used.

The FDTD method classically operates on structured Cartesian grids [23]. In general, the FIT can be applied to a greater variety of grids. However, in many cases grids are used which easily allow the generation of dual-orthogonal grids such as structured Cartesian grids [85]. To approximate complex-shaped materials on structured Cartesian grids several additional approaches for the FIT and the FDTD method have been developed. The oldest and simplest additional approach that considers complex shaped materials on Cartesian meshes is the *staircasing approximation*. This approach considers grid elements intersected by a material interface or material boundary as elements which are completely filled with one particular material. Due to the inaccurate approximation of the material interfaces or material boundaries, it performs poorly with regard to accuracy. Therefore, further methods have been developed to overcome the accuracy shortcomings. Two of the most prominent examples are the *Partially Filled Cell* (PFC) approach introduced in [80] for FIT and the locally conformal boundary FDTD algorithm in [23]. Both techniques show a remarkable decrease in the approximation error on problems with complex shaped materials.

The advantage of the FIT and the FDTD-type methods is their simplicity [85] which leads to very efficient schemes in terms of computational time. However, both techniques are mainly used with lowest order approximations. High order FDTD or FIT methods usually rely on a large spatial stencil which makes the treatment of curved material boundaries extremely cumbersome and numerically inefficient (see, e.g. [76]).

The *Finite Element* (FE) method has been first introduced in 1943 in [22]. In the 1950s the method was further developed by [5] and [83] in the field of civil engineering where it was used to solve mechanical applications. Since then, the method has become well established and has been used successfully in many fields of engineering. In the field of electrical engineering the first papers on this method were published in 1970 (see e.g. [72]). In contrast to the FIT and the FDTD method, the FE method is based on a variational or weak formulation of the Maxwell's equations [60]. Furthermore, the solution quantities are approximated using polynomial shape functions with a local support spanning the so-called local finite element spaces [12].

In contrast to the FIT and the FDTD methods, the FE method can be easily implemented with high order approximations. Furthermore, curved conforming meshes can be applied to problems including arbitrarily shaped materials which fit the material interface or boundary exactly. These meshes substantially reduce the FE approximation error on arbitrarily shaped materials (see, e.g. [13], [12]).

However, when considering moving material distributions, the FE method needs a remeshing or mesh adaption scheme to adapt the conforming mesh to the material geometry in the subsequent point in time. For certain applications the meshing can become a very challenging and cumbersome task. For instance, certain geometries that are obtained from measurements or complex Computer-Aided Design (CAD) models are difficult to mesh [57]. Another example are biomedical applications which require the meshing of very complex objects such as organs or blood vessels [57]. We refer to [75], [15] and [57] for further information on this topic.

To avoid the often computational costly meshing or mesh adaption techniques, we follow an alternative approach. The main idea is to use higher order approximations on simple and efficient hexahedral meshes and perform an additional approach to consider complex curved material domains. This approach is inspired both by the above mentioned PFC approach for FIT [80] and the locally conformal FDTD algorithm [23].

To implement this idea, it turns out that the DG method is well suited for the following reasons: First, it is based on a variational or weak formulation of the Maxwell's equations. Second, high order approximations can be easily implemented. Third, in contrast to the FE method, the DG method offers a greater flexibility for additional approaches since no continuity constraints have to be satisfied at the element interfaces. In the next section, we will present the DG method in greater detail. For further information on the FDTD-type method, FIT and FE method, the reader is referred to literature (see, e.g. [92], [89] [88], [13], [12], [85])

3.2 Discontinuous Galerkin Method

The *Discontinuous Galerkin* (DG) method has been introduced in [67] in 1973 where it was applied to steady-state neutron transport equations. Since then it has been further developed for various applications. Well-known DG methods are, among others, the DG method of Bassi and Rebay in [8], the Local DG (LDG) method in [20] or the discontinuous hp method of Baumann and Oden in [10]. In the field of electromagnetics, several DG methods have been proposed for the full system of the Maxwell's equations. A review of DG methods can be found in [38], [21], [39] and [33].

Note that the DG methods have been presented in various ways. Several DG methods apply the DG discretization only in time and a continuous finite element dicretization in space such as the DG methods in [27], [46] and [56]. Therefore, we would like to point out that we present the DG method the way it was introduced in [20]. In [20], the DG method is applied to the spatial derivatives only and not to the time derivative of the equations.

The DG method which is presented below is characterized by the following features: First, the method is based on a variation or weak formulation of the Maxwell's equations. Second, the solution quantities can be approximated using high order shape functions which span local finite element spaces. Furthermore, the elements in the mesh are connected to each other using numerical flux terms which act at the element interfaces as described in Section 3.2.3. In contrast to the FE method, this implies that no continuity conditions have to be satisfied at element interfaces [20]. The non-existing continuity constraints provide a greater flexibility [18]. In our case, it naturally enables the implementation of additional approaches which are embedded in the DG method such as the boundary conformal approach which is presented in Section 3.3. However, the advantage of a greater flexibility comes at the cost of slightly higher amount of degrees of freedom (DOF). A more detailed description of the properties, advantages and drawbacks of the DG method is given in Section 3.2.6. Furthermore, a method to lower the high number of DOF is presented in Section 3.3.5.

Finally, we would like to state that the DG method is best described as a combination of both the well-known *Finite Volume* (FV) method and the FE method, because the DG method uses a variational formulation similar to the FE method and, furthermore, numerical flux terms which show remarkable similarities to the FV method [18].

3.2.1 Computational domain and approximation spaces

In the following we are given a computational domain $\Omega \subset \mathbb{R}^3$ with Lipschitz continuous boundaries $\Gamma = \partial\Omega$. Furthermore, we assume to have a regular partition of the computational domain into smaller domains which are usually referred to as elements. The partition of the computational domain into elements is called mesh and defined by

$$\Omega_h = \{K_i\}_{i=1,\dots,N}, \tag{3.1}$$

where K_i represents an element and N the number of elements of the mesh. Furthermore, the mesh has to satisfy the following properties (see e.g. [13], [93]):

- all elements are non-overlapping, i.e. $\mathrm{int}(K_i) \cap \mathrm{int}(K_j) = \emptyset$ for $i \neq j$
- the elements completely cover the computational domain, i.e. $\Omega = \cup_{K_i \in \Omega_h} K_i$
- the intersection of two elements, i.e. $K_i \cap K_j$ for $i \neq j$, are either faces, edges or vertices of both elements or the empty set.

The full definition of a Lipschitz continuous boundary can be found in [60]. The cone property which is strongly linked to the Lipschitz boundary condition is introduced in [41]. We refer to [40] for examples of very simple shaped domains with boundaries that are not considered as Lipschitz boundaries.

Computational domains with Lipschitz continuous boundaries are advantageous for several reasons: First, such domains can always be covered by a mesh [60]. Second, they allow for an easier mathematical representation of the DG method [60]. Furthermore, the unit outward normal vector $\mathbf{n}$ is well-defined at almost every point on Γ [60].

In most applications the elements of a mesh are geometrically simple shaped objects. The most commonly used geometrical objects are quadrilateral, triangular, tetrahedral and hexahedral elements [42]. The DG method presented in Section 3.3 is tailor-made for a partition of the computational domain into hexahedral non-overlapping elements. Therefore, we restrict ourselves to hexahedral elements. Nevertheless, the method can be modified easily so that it can be applied to other types of element.

The boundary Γ of the computational domain Ω is furthermore partitioned into a boundary with Neumann boundary conditions Γ_N and into a boundary with Dirichlet boundary conditions Γ_D. Moreover, the following notations are used:

$\mathcal{N} = \{n_i\}$	set of nodes,	$\mathcal{E} = \{e_i\}$	set of edges,
$\mathcal{F} = \{f_i\}$	set of faces,	$\Omega_h = \{K_i\}$	mesh or computational domain.

Furthermore, let $\mathcal{F}_N$ and $\mathcal{F}_D$ denote the set of boundary faces with Neumann and Dirichlet boundary conditions, respectively, and $\mathcal{F}_0$ the set of interior faces. The parameter h refers to the minimum element length of the hexahedral elements and is referred to as grid length in the following.

3.2.2 Semi-discrete weak formulation

In this section, we derive the semi-discrete weak DG formulation of the Maxwell's equations. We follow the standard Local Discontinuous Galerkin (LDG) approach for second order partial differential equations (see e.g. [38] and [20]). Therefore, the DG method is applied to the *first-order formulation* - also known as *mixed formulation* - of the Maxwell's equations which reads

$$\frac{1}{\epsilon}\mathbf{D}(\mathbf{x},t) = -\nabla\phi(\mathbf{x},t), \tag{3.2}$$

$$\nabla\cdot\left(\frac{\partial}{\partial t}\mathbf{D}(\mathbf{x},t) + \frac{\kappa(\mathbf{x})}{\epsilon(\mathbf{x})}\mathbf{D}(\mathbf{x},t)\right) = 0, \tag{3.3}$$

$\forall t \in (0,T)$ and $\forall \mathbf{x} \in \Omega$ with the initial and boundary conditions

$$\begin{aligned} \phi(\mathbf{x}, t=0) &= \phi_0 && \forall \mathbf{x} \text{ in } \Omega, && (3.4)\\ \phi(\mathbf{x}, t) &= \phi_D && \forall \mathbf{x} \text{ on } \Gamma_D, && (3.5)\\ \mathbf{D}(\mathbf{x}, t) &= \mathbf{0} && \forall \mathbf{x} \text{ on } \Gamma_N. && (3.6) \end{aligned}$$

Note that we restrict ourselves to homogeneous Neumann boundary conditions in the following. Next, we have to define the approximate solutions of the electric potential and the flux density, denoted by $\phi_h(\mathbf{x}, t)$ and $\mathbf{D}_h(\mathbf{x}, t)$, which we need for the introduction of the semi-discrete weak formulation. The approximate solutions, $\phi_h(\mathbf{x}, t)$ and $\mathbf{D}_h(\mathbf{x}, t)$, belong to the finite element spaces

$$\begin{aligned} V_h &= \{v \in L^2(\Omega) : v|_{K_i} \in \mathcal{S}(K_i)\ \forall K_i \in \Omega_h\}, && (3.7)\\ \Sigma_h &= \{\boldsymbol{\tau} \in \left[L^2(\Omega)\right]^3 : \boldsymbol{\tau}|_{K_i} \in \mathcal{U}(K_i) \forall K_i \in \Omega_h\}, && (3.8) \end{aligned}$$

respectively, where $\mathcal{S}(K_i) := \mathcal{P}^p(K_i)$ and $\mathcal{U}(K_i) := [\mathcal{P}^p(K_i)]^3$ represent the local finite element spaces. Both spaces are polynomial spaces defined on each Cartesian element K_i and at most of order $p \in \mathbb{N}$. Furthermore, the approximate solutions $\phi_h(\mathbf{x}, t)$ and $\mathbf{D}_h(\mathbf{x}, t)$ can be defined as the direct sum over all piecewise p-th order polynomial approximations $\phi_h(\mathbf{x}, t)|_{K_i \in \Omega_h} = \phi_h^{K_i}(\mathbf{x}, t)$ and $\mathbf{D}_h(\mathbf{x}, t)|_{K_i \in \Omega_h} = \mathbf{D}_h^{K_i}(\mathbf{x}, t)$:

$$\phi_h(\mathbf{x}, t) = \bigoplus_{i=1}^{N} \phi_h^{K_i}(\mathbf{x}, t), \tag{3.9}$$

$$\mathbf{D}_h(\mathbf{x}, t) = \bigoplus_{i=1}^{N} \mathbf{D}_h^{K_i}(\mathbf{x}, t), \tag{3.10}$$

where the piecewise polynomial approximations on each Cartesian element K_i can be expressed in terms of basis functions:

$$\phi_h^{K_i}(\mathbf{x}, t) = \sum_{i=1}^{N^p} \phi_h^i(t) v_i(\mathbf{x}), \tag{3.11}$$

$$\mathbf{D}_h^{K_i}(\mathbf{x}, t) = \sum_{i=1}^{N^p} D_h^i(t) \boldsymbol{\tau}_i(\mathbf{x}). \tag{3.12}$$

After having defined the approximate solutions ϕ_h and $\mathbf{D}_h$, we can derive the weak DG formulation of (3.2)-(3.3) to completely specify the DG method. Following [20] and [7], we multiply (3.2)-(3.3) by the scalar and vector test functions denoted as v and $\boldsymbol{\tau}$, respectively, and integrate both equations over the computational domain Ω_h. The DG method as introduced in [20] and [7] is based on the Galerkin scheme. This implies that the space spanned by the test functions and the space spanned by the basis functions are identical, i.e. $v \in V_h$ and $\boldsymbol{\tau} \in \Sigma_h$.
Next, the Gauss' theorem is applied to (3.2)-(3.3). Furthermore, introducing the numerical flux terms, $\tilde{\phi}_h$ and $\tilde{\mathbf{D}}_h$, which uniquely define the values at the element interface, leads to the semi-

discrete weak DG flux formulation:
Find $\mathbf{D}_h(\mathbf{x},t)$, $\phi_h(\mathbf{x},t)$ such that $\forall K_i \in \Omega_h$

$$\begin{aligned} \int_{K_i} \boldsymbol{\tau} \cdot \frac{1}{\epsilon} \mathbf{D}_h dV &= \int_{K_i} \nabla \cdot \boldsymbol{\tau} \phi_h dV - \int_{\partial K_i} \tilde{\phi}_h \boldsymbol{\tau} \cdot \mathbf{n} ds \qquad \forall \boldsymbol{\tau} \in \Sigma_h, \quad (3.13) \\ \int_{K_i} \nabla \upsilon \cdot \left(\frac{\partial}{\partial t} \mathbf{D}_h + \frac{\kappa}{\epsilon} \mathbf{D}_h \right) dV &= \int_{\partial K_i} \upsilon \left(\frac{\partial}{\partial t} \tilde{\mathbf{D}}_h + \frac{\kappa}{\epsilon} \tilde{\mathbf{D}}_h \right) \cdot \mathbf{n} ds \qquad \forall \upsilon \in V_h, \quad (3.14) \end{aligned}$$

where $\mathbf{n}$ denotes the outward pointing interface normal. The numerical flux terms, $\tilde{\phi}_h$ and $\tilde{\mathbf{D}}_h$, are suitable approximations for the discontinuous solutions ϕ_h and $\mathbf{D}_h$ at the element interfaces ∂K_i. The definition of the numerical flux terms $\tilde{\mathbf{D}}_h$ and $\tilde{\phi}_h$ is presented in the next section.

3.2.3 Numerical flux formulation

The numerical fluxes are discrete approximations to the traces at the element boundaries. They are crucial as they connect the elements to each other and, therefore, directly affect consistency, stability and accuracy of the method [38]. Clearly, they also have an impact on the sparsity and symmetry of the resulting matrices [18]. In this work, the numerical flux formulation of the *Local Discontinuous Galerkin* (LDG) method as introduced in [20] is considered. For the sake of simplicity, we define the fluxes using the average and jump notation of [7] at the element interfaces: Let $f_i \in \mathcal{F}_0$ be a particular interior face which is shared by two adjacent elements, here denoted as K^+ and K^-. The average and jump of a scalar function u with $u^\pm := u|_{\partial K^\pm}$ is defined by

$$\begin{aligned} \{u\} &= \frac{1}{2}\left(u^+ + u^-\right) \qquad \text{on } f_i \in \mathcal{F}_0, \quad (3.15) \\ [\![u]\!] &= u^+ \mathbf{n}^+ + u^- \mathbf{n}^- \qquad \text{on } f_i \in \mathcal{F}_0, \quad (3.16) \end{aligned}$$

For a vector function $\mathbf{q}$ with $\mathbf{q}^\pm := \mathbf{q}|_{\partial K^\pm}$ the average $\{\mathbf{q}\}$ and jump $[\![\mathbf{q}]\!]$ are defined as

$$\begin{aligned} \{\mathbf{q}\} &= \frac{1}{2}\left(\mathbf{q}^+ + \mathbf{q}^-\right) \qquad \text{on } f_i \in \mathcal{F}_0, \quad (3.17) \\ [\![\mathbf{q}]\!] &= \mathbf{q}^+ \cdot \mathbf{n}^+ + \mathbf{q}^- \cdot \mathbf{n}^- \qquad \text{on } f_i \in \mathcal{F}_0, \quad (3.18) \end{aligned}$$

respectively, where $\mathbf{n}^+$ and $\mathbf{n}^-$ refer to the associated outward pointing unit normals of K^+ and K^-, respectively.
Using this notation, the numerical flux terms in (3.7)-(3.8) for all interior faces are defined as

$$\begin{aligned} \tilde{\phi}_h &= \{\phi_h\} + \mathbf{C}_{12} \cdot [\![\phi_h]\!] \qquad \text{on } f_i \in \mathcal{F}_0, \quad (3.19) \\ \tilde{\mathbf{D}}_h &= \{\mathbf{D}_h\} - \mathbf{C}_{12} [\![\mathbf{D}_h]\!] - C_{11} [\![\phi_h]\!] \qquad \text{on } f_i \in \mathcal{F}_0, \quad (3.20) \end{aligned}$$

where $C_{11} \in \mathbb{R}$ and $\mathbf{C}_{12} \in \mathbb{R}^{3\times 3}$ are stabilization parameters which are set to a constant value on each face. The stabilization parameters can be either interpreted as penalization terms of the jumps or as damping terms that stabilize the DG method [18].

Since the boundary conditions are weakly imposed by the numerical fluxes, the numerical flux formulation on boundary faces with Dirichlet boundary conditions reads

$$\begin{aligned}\tilde{\phi}_h &= \phi_D \quad \text{on } f_i \in \mathcal{F}_D, && (3.21)\\ \tilde{\mathbf{D}}_h &= \mathbf{D}_h^+ - C_{11}\left(\phi_h^+ - \phi_D\right)\mathbf{n} \quad \text{on } f_i \in \mathcal{F}_D. && (3.22)\end{aligned}$$

On the boundary faces with Neumann boundary conditions we have

$$\begin{aligned}\tilde{\phi}_h &= \phi_h^+ \quad \text{on } f_i \in \mathcal{F}_N, && (3.23)\\ \tilde{\mathbf{D}}_h &= \mathbf{D}_N \quad \text{on } f_i \in \mathcal{F}_N. && (3.24)\end{aligned}$$

In the next section, we will introduce the weak DG formulation for the EQS and the ES approximation in bilinear form as well as in matrix formulation. We introduce both the bilinear form and the matrix formulation, since they enable us to write the weak DG formulation in a more compact form.

3.2.4 The mixed EQS setting - Bilinear form and matrix formulation

Summing (3.13)-(3.14) up over all elements and using the numerical fluxes as introduced above, the weak DG formulation can be written as (see [17]):
Find $\mathbf{D}_h(\mathbf{x},t)$, $\phi_h(\mathbf{x},t)$ such that

$$a\left(\frac{1}{\epsilon}\mathbf{D}_h, \boldsymbol{\tau}\right) = b\left(\phi_h, \boldsymbol{\tau}\right) + f\left(\boldsymbol{\tau}\right) \qquad \forall \boldsymbol{\tau} \in \Sigma_h, \quad (3.25)$$

$$-b\left(v, (\frac{\partial}{\partial t}\mathbf{D}_h + \frac{\kappa}{\epsilon}\mathbf{D}_h)\right) + s\left(v, (\frac{\partial}{\partial t}\phi_h + \frac{\kappa}{\epsilon}\phi_h)\right) = g(v) \qquad \forall v \in V_h, \quad (3.26)$$

with

$$\begin{aligned}a(\mathbf{D},\boldsymbol{\tau}) &:= \int_{\Omega_h} \boldsymbol{\tau}\cdot\mathbf{D}\,dV, && (3.27)\\ b(\phi,\boldsymbol{\tau}) &:= \sum_{K_i\in\Omega_h}\int_{K_i} \nabla\cdot\boldsymbol{\tau}\phi\,dV - \int_{\mathcal{F}_0}\left(\{\phi\} + \mathbf{C}_{12}\cdot[\![\phi]\!]\right)[\![\boldsymbol{\tau}]\!]\,dA, && (3.28)\\ s(v,\phi) &:= \int_{\mathcal{F}_0} C_{11}[\![\phi]\!]\cdot[\![v]\!]\,dA + \int_{\mathcal{F}_D} C_{11} v\phi\,dA, && (3.29)\\ f(\boldsymbol{\tau}) &:= \int_{\mathcal{F}_D} \phi_D\boldsymbol{\tau}\cdot\mathbf{n}\,dA, && (3.30)\\ g(v) &:= \int_{\mathcal{F}_D} C_{11}\left(\frac{\partial}{\partial t}\phi_D + \frac{\kappa}{\epsilon}\phi_D\right) v\,dA, && (3.31)\end{aligned}$$

Using the Gauss theorem the term $b(\phi,\boldsymbol{\tau})$ can also be written as (see e.g. [7])

$$b(\phi,\boldsymbol{\tau}) = -\sum_{K_i\in\Omega_h}\int_{K_i}\nabla\phi\cdot\boldsymbol{\tau}\,dV + \int_{\mathcal{F}_0}\left(\{\boldsymbol{\tau}\} - \mathbf{C}_{12}\cdot[\![\boldsymbol{\tau}]\!]\right)[\![\phi]\!]\,dA + \int_{\mathcal{F}_D}\phi\boldsymbol{\tau}\cdot\mathbf{n}\,dA. \quad (3.32)$$

Moreover, expressing the approximate solutions ϕ_h and $\mathbf{D}_h$ by means of the basis functions υ and $\boldsymbol{\tau}$ yields the following matrix representation:

$$\begin{aligned} \mathbf{M}\mathbf{I}_{1/\epsilon}\mathbf{d} &= \mathbf{G}\boldsymbol{\phi} + \mathbf{f}, & (3.33)\\ -\left(\mathbf{G}^T\frac{d}{dt}\mathbf{d} + \mathbf{G}^T\mathbf{I}_{\kappa/\epsilon}\mathbf{d}\right) + \left(\mathbf{S}\frac{d}{dt}\boldsymbol{\phi} + \mathbf{S}\mathbf{I}_{\kappa/\epsilon}\boldsymbol{\phi}\right) &= \mathbf{g}, & (3.34)\end{aligned}$$

where $\mathbf{d}$ and $\boldsymbol{\phi}$ denote the vectors of the potential and the flux density degrees of freedom, respectively. The terms $\mathbf{M}$, $\mathbf{G}$ and $\mathbf{S}$ represent the mass, gradient and stability term matrices, respectively, and are given by

$$\begin{aligned} \mathbf{M}_{ij} &:= \int_{\Omega_h} \boldsymbol{\tau}_i \cdot \boldsymbol{\tau}_j dV, & (3.35)\\ \mathbf{G}_{ij} &:= \sum_{K_i \in \Omega_h} \int_{K_i} \nabla \cdot \boldsymbol{\tau}_i \upsilon_j dV - \int_{\mathcal{F}_0} \left(\{\upsilon_j\} + \mathbf{C}_{12} \cdot [\![\upsilon_j]\!]\right) [\![\boldsymbol{\tau}_i]\!] dA, & (3.36)\\ \mathbf{S}_{ij} &:= \int_{\mathcal{F}_0} C_{11} [\![\upsilon_j]\!] [\![\upsilon_i]\!] dA + \int_{\mathcal{F}_D} C_{11} \upsilon_j \boldsymbol{\tau}_i \cdot \mathbf{n} dA, & (3.37)\end{aligned}$$

Finally, we introduce the vectors $\mathbf{f}$ and $\mathbf{g}$ which contain all boundary condition terms. These vectors are defined as follows:

$$\begin{aligned} \mathbf{f}_i &:= \int_{\mathcal{F}_D} \phi_D \boldsymbol{\tau}_i \cdot \mathbf{n} dA, & (3.38)\\ \mathbf{g}_i &:= \int_{\mathcal{F}_D} C_{11} \left(\frac{\partial}{\partial t}\phi_D + \frac{\kappa}{\epsilon}\phi_D\right) \upsilon_i dA. & (3.39)\end{aligned}$$

The terms $\mathbf{I}_{1/\epsilon}$ and $\mathbf{I}_{\kappa/\epsilon}$ are diagonal matrices with the cell-wise constant material parameters on the diagonal.

The LDG flux formulation has the advantage that the numerical flux term of the electric potential is free of terms which depend on the electric flux density. This implies that the mass operator $\mathbf{M}$ in the DG formulation is block-diagonal and can easily be inverted. Therefore, it is straightforward to apply the *Schur complement* reduction in (3.33)-(3.34) which yields the semi-discrete system of equations

$$\left(-\mathbf{G}^T\mathbf{I}_\epsilon\mathbf{M}^{-1}\mathbf{G} + \mathbf{S}\right)\frac{d}{dt}\boldsymbol{\phi} = \left(\mathbf{G}^T\mathbf{I}_\kappa\mathbf{M}^{-1}\mathbf{G} - \mathbf{S}\mathbf{I}_{\kappa/\epsilon}\right)\boldsymbol{\phi} + \mathbf{g}_B. \qquad (3.40)$$

where $\mathbf{g}_B = \mathbf{M}^{-1}\mathbf{I}_\epsilon\mathbf{f} + \mathbf{g}$ denotes the vector of boundary conditions. Note that instead of solving (3.33)-(3.34), we solve (3.40) for the potential solution $\boldsymbol{\phi}$. Furthermore, note that the flux density solution $\mathbf{d}$ is eliminated from (3.40). However, it can be easily obtained by plugging the potential solution $\boldsymbol{\phi}$ into (3.33).

3.2.5 The mixed ES setting - Bilinear form and matrix formulation

In Section 3.2.4, we have introduced the bilinear form and matrix formulation of the discrete weak DG formulation for the EQS approximation of the Maxwell's equation. Note that the discrete weak DG formulation of the ES approximation of the Maxwell's equation can be derived in exactly the same way. For this reason, we only present the bilinear form and the matrix formulation of the ES approximation without deriving them explicitly. The reader is referred to Section 3.2.2, 3.2.3 and 3.2.4 or [17] for details on the derivation of the discrete weak DG formulation of the ES approximation.

The bilinear form of discrete weak DG formulation for the ES approximation of the Maxwell's equations reads (see, e.g. [17]):

Find $\mathbf{D}_h(\mathbf{x}, t)$, $\phi_h(\mathbf{x}, t)$ such that

$$a\left(\frac{1}{\epsilon}\mathbf{D}_h, \boldsymbol{\tau}\right) = b\left(\phi_h, \boldsymbol{\tau}\right) + f\left(\boldsymbol{\tau}\right) \qquad \forall \boldsymbol{\tau} \in \Sigma_h, \qquad (3.41)$$

$$-b\left(\upsilon, \mathbf{D}_h\right) + s\left(\upsilon, \phi_h\right) = g(\upsilon) \qquad \forall \upsilon \in V_h, \qquad (3.42)$$

with $a(\mathbf{D}, \boldsymbol{\tau})$, $b(\phi, \boldsymbol{\tau})$, $s(\upsilon, \phi)$ and $f(\boldsymbol{\tau})$ defined as shown in (3.27), (3.28), (3.29) and (3.30), respectively. Note that the linear form $g(\upsilon)$ is defined differently for the ES approximation. It is given by

$$g(\upsilon) \quad := \quad -\int_{\Omega_H} \rho \upsilon dV + \int_{\mathcal{F}_D} C_{11}\phi_D \upsilon \cdot \mathbf{n} dA. \qquad (3.43)$$

The matrix formulation of the ES approximation is written as follows:

$$\mathbf{M}\mathbf{I}_{1/\epsilon}\mathbf{d} \quad = \quad \mathbf{G}\boldsymbol{\phi} + \mathbf{f}, \qquad (3.44)$$

$$-\mathbf{G}^T\mathbf{d} + \mathbf{S}\boldsymbol{\phi} \quad = \quad \mathbf{g}, \qquad (3.45)$$

where the matrices $\mathbf{M}$, $\mathbf{G}$ and $\mathbf{S}$ and the vector $\mathbf{f}$ are defined as presented in (3.35), (3.36), (3.37) and (3.38), respectively. However, the vector $\mathbf{g}$ is given as

$$\mathbf{g}_i \quad := \quad -\int_{\Omega_h} \upsilon_i \rho dV + \int_{\mathcal{F}_D} C_{11}\phi_D \upsilon_i dA. \qquad (3.46)$$

As for the EQS approximation, a *Schur complement* reduction can be applied to (3.44)-(3.45). Therefore, the matrix formulation in (3.44)-(3.45) reduces to

$$\left(-\mathbf{G}^T\mathbf{I}_\epsilon\mathbf{M}^{-1}\mathbf{G} + \mathbf{S}\right)\boldsymbol{\phi} = \mathbf{g}_B. \qquad (3.47)$$

where $\mathbf{g}_B$ is given by $\mathbf{g}_B = \mathbf{M}^{-1}\mathbf{I}_\epsilon\mathbf{f} + \mathbf{g}$. The potential solution $\boldsymbol{\phi}$ is obtained by solving (3.47) for $\boldsymbol{\phi}$. Note that the solvers which are applied to the matrix formulation in (3.47) will be presented in Section 3.5. The flux density solution $\mathbf{d}$ can be evaluated by entering the potential solution $\boldsymbol{\phi}$ in (3.44).

3.2.6 Numerical properties of the Discontinuous Galerkin method

This section focuses on the main properties of the Discontinuous Galerkin method. The most important properties of numerical methods such as the DG method are stability, consistency, conservativeness and convergence [38]. We will show that all four properties are satisfied for the EQS approximation of the Maxwell's equations. Note that the properties are also satisfied for the ES approximations. However, since this was shown in many papers of other authors, we've decided to focus on the EQS approximation of the Maxwell's equations in this section. The reader is referred to the literature for further details on the properties for the EQS approximation of the Maxwell's equations (see, e.g. [18], [17], [7]).

As presented in [18] the consistency of the DG method applied to the EQS approximation of the Maxwell's equations can be shown by replacing the approximate solutions ϕ_h and $\mathbf{D}_h$ in (3.13)-(3.14) by the exact solutions ϕ and $\mathbf{D}$ of (3.2)-(3.3). The method is consistent if the numerical fluxes satisfy $\tilde{\phi} = \phi$ and $\tilde{\mathbf{D}} = \mathbf{D}$, which is the case for the LDG numerical flux formulation as $\{\phi\} = \phi$, $\{\mathbf{D}\} = \mathbf{D}$, $[\![\phi]\!] = 0$ and $[\![\mathbf{D}]\!] = 0$.

Furthermore, it can be shown that the method is local conservative if the vector flux $\tilde{\mathbf{D}}_h$ is single-valued. And indeed, this is the case for the chosen LDG flux formulation (see, e.g. [7]). Therefore, if the scalar test function v in (3.14) is set to 1, the charge conservation law for any given union of elements $S = \cup_{i=1,\dots,j} K_i$ in (2.11) holds

$$\int_{\partial S} \left(\frac{\partial}{\partial t} \tilde{\mathbf{D}}_h + \frac{\kappa}{\epsilon} \tilde{\mathbf{D}}_h \right) \cdot \mathbf{n} dA = 0. \tag{3.48}$$

Next, we will present the stability proof for the DG method where we follow [18] and restrict ourselves to homogeneous Dirichlet boundary conditions. The stability proof for inhomogeneous Dirichlet boundary conditions is an extension of the proof shown below (see, e.g. [20] or [17]).

Before we can show stability of the numerical scheme, the stability result of the mixed formulation in (3.2)-(3.3) has to be derived. Therefore, we first multiply (3.2)-(3.3) by the exact solution $\mathbf{D}$ and ϕ and integrate over the computational domain. This yields

$$\int_\Omega \frac{1}{\epsilon} \mathbf{D} \cdot \mathbf{D} dV = - \int_\Omega \nabla \phi \cdot \mathbf{D} dV, \tag{3.49}$$

and

$$- \int_\Omega \left(\frac{\partial}{\partial t} \mathbf{D} + \frac{\kappa}{\epsilon} \mathbf{D} \right) \cdot \nabla \phi dV = 0. \tag{3.50}$$

Multiplying (3.50) by $\frac{\epsilon}{\kappa}$ and adding (3.49) to (3.50) yields

$$- \int_\Omega \frac{\epsilon}{\kappa} \frac{\partial}{\partial t} \mathbf{D} \cdot \nabla \phi dV + \int_\Omega \frac{1}{\epsilon} \mathbf{D} \cdot \mathbf{D} dV = 0, \tag{3.51}$$

what can be expressed as

$$- \int_\Omega \epsilon \frac{\partial}{\partial t} \mathbf{D} \cdot \nabla \phi dV + \int_\Omega \frac{\kappa}{\epsilon} \mathbf{D} \cdot \mathbf{D} dV = 0. \tag{3.52}$$

Since we have $-\nabla \phi = \frac{1}{\epsilon} \mathbf{D}$, the stability result of the mixed formulation in (3.2)-(3.3) is given by

$$\frac{d}{dt} \int_\Omega \frac{1}{2} |\mathbf{D}|^2 dV + \int_\Omega \frac{\kappa}{\epsilon} |\mathbf{D}|^2 dV = 0. \tag{3.53}$$

To show stability of the DG approach we follow a similar procedure applied to the weak DG formulation in (3.13)-(3.14). Summing (3.13) and (3.14) up over all elements and entering $\boldsymbol{\tau} = \mathbf{D}_h$ into (3.13) results in

$$\int_{\Omega_h} \mathbf{D}_h \cdot \frac{1}{\epsilon}\mathbf{D}_h dV = \sum_{K_i \in \Omega_h} \left(\int_{K_i} \nabla \cdot \mathbf{D}_h \phi_h dV - \int_{\partial K_i} \tilde{\phi}_h \mathbf{D}_h \cdot \mathbf{n} dA \right). \tag{3.54}$$

Furthermore, using $v = \phi_h$ in (3.14) yields

$$\sum_{K_i \in \Omega_h} \left(- \int_{K_i} \left(\nabla \phi_h \cdot \frac{\partial}{\partial t}\mathbf{D}_h + \nabla \phi_h \cdot \frac{\kappa}{\epsilon}\mathbf{D}_h \right) dV + \int_{\partial K_i} \left(\phi_h \frac{\partial}{\partial t}\tilde{\mathbf{D}}_h \cdot \mathbf{n} + \phi_h \frac{\kappa}{\epsilon}\tilde{\mathbf{D}}_h \cdot \mathbf{n} \right) dA \right) = 0. \tag{3.55}$$

Multiplying (3.55) by $\frac{\epsilon}{\kappa}$ and summing (3.54) and (3.55), we obtain

$$\sum_{K_i \in \Omega_h} \left(- \int_{K_i} \nabla \phi_h \cdot \frac{\epsilon}{\kappa}\frac{\partial}{\partial t}\mathbf{D}_h dV + \int_{\partial K_i} \phi_h \frac{\epsilon}{\kappa}\frac{\partial}{\partial t}\tilde{\mathbf{D}}_h \cdot \mathbf{n} dA \right) + \int_{\Omega_h} \mathbf{D}_h \cdot \frac{1}{\epsilon}\mathbf{D}_h dV + \Theta_h = 0, \tag{3.56}$$

where

$$\begin{aligned} \Theta_h &= \sum_{K_i \in \Omega_h} \left(- \int_{K_i} \left(\nabla \cdot \mathbf{D}_h \phi_h + \nabla \phi_h \cdot \mathbf{D}_h \right) dV + \int_{\partial K_i} \left(\tilde{\phi}_h \mathbf{D}_h \cdot \mathbf{n} + \phi_h \tilde{\mathbf{D}}_h \cdot \mathbf{n} \right) dA \right) \\ &= \sum_{K_i \in \Omega_h} \left(\int_{K_i} -\nabla \cdot \left(\mathbf{D}_h \phi_h \right) dV + \int_{\partial K_i} \left(\tilde{\phi}_h \mathbf{D}_h + \phi_h \tilde{\mathbf{D}}_h \right) \cdot \mathbf{n} dA \right) \\ &= \sum_{K_i \in \Omega_h} \int_{\partial K_i} \left(-\mathbf{D}_h \phi_h + \tilde{\phi}_h \mathbf{D}_h + \phi_h \tilde{\mathbf{D}}_h \right) \cdot \mathbf{n} dA. \end{aligned} \tag{3.57}$$

Remember that the approximate solution $\mathbf{D}_h$ is defined as $\mathbf{D}_h = \sum_{K_i} \sum_i D_h^i(t) \boldsymbol{\tau}_i$. Since the approximation functions $\boldsymbol{\tau}$ do not depend on the time variable, equation (3.54) is also satisfied for the time derivative $\frac{\partial}{\partial t}\mathbf{D}_h$. Therefore, we have

$$\begin{aligned} \int_{\Omega_h} \mathbf{D}_h \cdot \frac{1}{\epsilon}\frac{\partial}{\partial t}\mathbf{D}_h dV &= \sum_{K_i \in \Omega_h} \left(\int_{K_i} \nabla \cdot \frac{\partial}{\partial t}\mathbf{D}_h \phi_h dV - \int_{\partial K_i} \tilde{\phi}_h \frac{\partial}{\partial t}\mathbf{D}_h \cdot \mathbf{n} dA \right) \\ &= \sum_{K_i \in \Omega_h} \left(- \int_{K_i} \nabla \phi_h \cdot \frac{\partial}{\partial t}\mathbf{D}_h dV + \int_{\partial K_i} \phi_h \frac{\partial}{\partial t}\tilde{\mathbf{D}}_h \cdot \mathbf{n} dA \right) - \Xi_h \end{aligned} \tag{3.58}$$

with

$$\Xi_h = \sum_{K_i \in \Omega_h} \int_{\partial K_i} \left(-\phi_h \frac{\partial}{\partial t}\mathbf{D}_h + \phi_h \frac{\partial}{\partial t}\tilde{\mathbf{D}}_h + \tilde{\phi}_h \frac{\partial}{\partial t}\mathbf{D}_h \right) \cdot \mathbf{n} dA. \tag{3.59}$$

Using (3.58) and the relation $\int_{\Omega_h} \mathbf{D}_h \cdot \frac{1}{\epsilon}\frac{\partial}{\partial t}\mathbf{D}_h dV = \frac{d}{dt}\int_{\Omega_h} \frac{1}{2\epsilon}|\mathbf{D}_h|^2 dV$, equation (3.56) can be written as

$$\frac{d}{dt}\int_{\Omega_h} \frac{1}{2}|\mathbf{D}_h|^2 dV + \int_{\Omega_h} \mathbf{D}_h \cdot \frac{\kappa}{\epsilon}\mathbf{D}_h dV + \kappa\Theta_h + \epsilon\Xi_h = 0. \tag{3.60}$$

To show stability of the method, it essentially remains to show that the terms Θ_H and Ξ_H are non-negative.

Using the jump and average notation of Section 3.2.3, the terms Θ_h and Ξ_H can be written as

$$\begin{aligned}
\Theta_h =& \sum_{K_i} \int_{\partial K_i} \left(-\mathbf{D}_h \phi_h + \tilde{\phi}_h \mathbf{D}_h + \phi_h \tilde{\mathbf{D}}_h\right) \cdot \mathbf{n} dA \qquad (3.61)\\
=& \int_{\mathcal{F}} \llbracket -\mathbf{D}_h \phi_h + \tilde{\phi}_h \mathbf{D}_h + \phi_h \tilde{\mathbf{D}}_h \rrbracket dA \\
=& \int_{\mathcal{F}_0} \llbracket -\mathbf{D}_h \phi_h \rrbracket + \tilde{\phi}_h \llbracket \mathbf{D}_h \rrbracket + \llbracket \phi_h \rrbracket \cdot \tilde{\mathbf{D}}_h dA + \int_{\partial\Omega_h} \left(-\mathbf{D}_h \phi_h + \tilde{\phi}_h \mathbf{D}_h + \phi_h \tilde{\mathbf{D}}_h\right) \cdot \mathbf{n} dA \\
=& \int_{\mathcal{F}_0} \left(\left(\tilde{\phi}_h - \{\phi_h\}\right) \llbracket \mathbf{D}_h \rrbracket + \llbracket \phi_h \rrbracket \cdot \left(\tilde{\mathbf{D}}_h - \{\mathbf{D}_h\}\right)\right) dA + \int_{\partial\Omega_h} \left(\phi_h \left(\tilde{\mathbf{D}}_h - \mathbf{D}_h\right) + \tilde{\phi}_h \mathbf{D}_h\right) \cdot \mathbf{n} dA,
\end{aligned}$$

$$\begin{aligned}
\Xi_h =& \sum_{K_i \in \Omega_h} \int_{\partial K_i} \left(-\phi_h \frac{\partial}{\partial t} \mathbf{D}_h + \phi_h \frac{\partial}{\partial t} \tilde{\mathbf{D}}_h + \tilde{\phi}_h \frac{\partial}{\partial t} \mathbf{D}_h\right) \cdot \mathbf{n} dA \qquad (3.62)\\
=& \int_{\mathcal{F}} \llbracket -\phi_h \frac{\partial}{\partial t} \mathbf{D}_h + \phi_h \frac{\partial}{\partial t} \tilde{\mathbf{D}}_h + \tilde{\phi}_h \frac{\partial}{\partial t} \mathbf{D}_h \rrbracket dA \\
=& \int_{\mathcal{F}_0} \llbracket -\frac{\partial}{\partial t} \mathbf{D}_h \phi_h \rrbracket + \tilde{\phi}_h \llbracket \frac{\partial}{\partial t} \mathbf{D}_h \rrbracket + \llbracket \phi_h \rrbracket \cdot \frac{\partial}{\partial t} \tilde{\mathbf{D}}_h dA + \\
& \int_{\partial\Omega_h} \left(-\frac{\partial}{\partial t} \mathbf{D}_h \phi_h + \tilde{\phi}_h \frac{\partial}{\partial t} \mathbf{D}_h + \phi_h \frac{\partial}{\partial t} \tilde{\mathbf{D}}_h\right) \cdot \mathbf{n} dA \\
=& \int_{\mathcal{F}_0} \left(\left(\tilde{\phi}_h - \{\phi_h\}\right) \llbracket \frac{\partial}{\partial t} \mathbf{D}_h \rrbracket + \llbracket \phi_h \rrbracket \cdot \left(\frac{\partial}{\partial t} \tilde{\mathbf{D}}_h - \{\frac{\partial}{\partial t} \mathbf{D}_h\}\right)\right) dA + \\
& \int_{\partial\Omega_h} \left(\phi_h \left(\frac{\partial}{\partial t} \tilde{\mathbf{D}}_h - \frac{\partial}{\partial t} \mathbf{D}_h\right) + \tilde{\phi}_h \frac{\partial}{\partial t} \mathbf{D}_h\right) \cdot \mathbf{n} dA.
\end{aligned}$$

If the numerical fluxes of the LDG flux formulation introduced in Section 3.2.3,

$$\begin{aligned}
\tilde{\phi}_h &= \{\phi_h\} + \mathbf{C}_{12} \cdot \llbracket \phi_h \rrbracket, && (3.63)\\
\tilde{\mathbf{D}}_h &= \{\mathbf{D}_h\} - \mathbf{C}_{12} \llbracket \mathbf{D}_h \rrbracket - C_{11} \llbracket \phi_h \rrbracket, && (3.64)
\end{aligned}$$

are entered in (3.61) and (3.62), we finally obtain the desired expression for Θ_h and Ξ_h:

$$\begin{aligned}
\Theta_h &= \sum_{f \in \mathcal{F}_0} \int_f -C_{11} \llbracket \phi_h \rrbracket^2 dA \geq 0 + \int_{\mathcal{F}_D} -C_{11} \phi_h^2 dA \geq 0, && (3.65)\\
\Xi_h &= \sum_{f \in \mathcal{F}_0} \int_f -C_{11} \llbracket \frac{\partial}{\partial t} \phi_h \rrbracket \cdot \llbracket \phi_h \rrbracket dA + \int_{\mathcal{F}_D} -C_{11} \phi_h \frac{\partial}{\partial t} \phi_h dA \geq 0. && (3.66)
\end{aligned}$$

This inequality holds if the parameter C_{11} is set to a negative value (see [18]). Therefore, in the following we assume that $C_{11} < 0$ is always satisfied. Finally, entering (3.65) and (3.66) in (3.60) shows stability of the DG method.

The existence and uniqueness of the DG method is directly linked to the compatibility condition which reads (see, e.g. [18] and [17]):

$$\phi_h \in V_h : \int_{K_i} \nabla \phi_h \boldsymbol{\tau} dV = 0 \qquad \forall \boldsymbol{\tau} \in \Sigma_h \quad \text{then} \quad \nabla \phi_h = \mathbf{0}. \tag{3.67}$$

In [17] it was shown that (3.67) holds if the local finite element spaces $\mathcal{S}(K_i)$ and $\mathcal{U}(K_i)$ for each element $K_i \in \Omega_h$ satisfy

$$\nabla \mathcal{S}(K_i) \subset \mathcal{U}(K_i), \tag{3.68}$$

which is obviously true in our case since the local finite element spaces are spanned by polynomial functions (see Section 3.2.7).
To ensure existence and uniqueness of the approximate solutions in (3.13)-(3.14), it is sufficient to show that the solution of (3.13)-(3.14) for homogeneous initial and boundary conditions is the trivial solution. Using equation (3.60), (3.65) and (3.66) we get

$$\frac{d}{dt} \int_{\Omega_h} \frac{1}{2} |\mathbf{D}_h|^2 dV + \int_{\Omega_h} \mathbf{D}_h \cdot \frac{\kappa}{\epsilon} \mathbf{D}_h dV + \kappa \Theta_h + \epsilon \Xi_h = 0. \tag{3.69}$$

Since we have $C_{11} < 0$, we can directly infer that (3.69) is satisfied if $\mathbf{D_h} = \mathbf{0}$, $[\![\phi_h]\!] = 0$ and $\phi_h = 0$ on $\mathcal{F}_D$ (see, e.g. [18]). Hence, using (3.69), equation (3.25) can be written as

$$\sum_{K_i \in \Omega_h} \int_{K_i} \nabla \phi \cdot \boldsymbol{\tau} dV = 0, \qquad \forall \boldsymbol{\tau} \in \Sigma_h. \tag{3.70}$$

Therefore, the existence and uniqueness of the approximate solution ϕ_h follows directly from the compatibility condition (3.67) which is satisfied for our choice of local finite element spaces. The convergence property of the DG method directly follows from the LAX equivalence theorem [51].

We have just shown that the four most important properties for numerical methods are satisfied for the DG method with a LDG flux formulation. We also have that the mass matrix $\mathbf{M}$ is block diagonal, which makes possible the application of the Schur complement in (3.33)-(3.34). Based on these observations, we decided to use the DG method with the LDG formulation as introduced in [20].
Moreover, in [17] and [20] the L^2-error estimate of the LDG method for the electrostatic approximation is derived and given by:

$$\left(\int_{\Omega_h} |\phi - \phi_h|^2 dV \right)^{1/2} \leq C h^{p+1} \qquad \text{and} \qquad \left(\int_{\Omega_h} |\mathbf{D} - \mathbf{D}_h|^2 dV + \Theta_h^2 \right)^{1/2} \leq C h^p, \tag{3.71}$$

if parameter C_{11} is set to $C_{11} = -c/h$ with $c \in \mathbb{R}^+$ and if the order which is required to fit the smoothness of the exact solutions $\mathbf{D}$ and ϕ is greater or equal than the approximation order p [17]. Note that the parameter $C \in \mathbb{R}$ depends on the regularity of ϕ and Ω. Our numerical results presented in Section 4.1 show the optimal convergence rates p and $p+1$. This is in line with the results other authors have reported regarding the optimal convergence rates (see, e.g [34]).

The fact that the DG method doesn't have to satisfy continuity conditions at the element interfaces has several key advantages. First, it makes the method highly parallelizable since the mass matrix is block diagonal and, therefore, easily invertible [38]. Second, the absence of continuity constraints implies that mesh refinement, mesh unrefinement and also irregular meshes with hanging nodes can be handled very easily [38]. In addition, the approximation order of the polynomials may differ across elements which makes the method highly attractive for hp-adaptivity [38]. Finally, the possibility to design problem depending fluxes allows for more flexibility and is a huge advantage compared to the standard finite element method.

The most notable drawback of the DG method is the larger number of DOF compared to the FEM method. In general, the higher number of DOF implies higher computational costs [38]. For this reason, the FE method might be more efficient for problems where the advantages of the DG method do not play a crucial role. This might be the case for EQS field problems with simple shaped non-moving material boundaries where the meshing of the geometry is not of major importance. In such cases, the use of the DG method might be less appealing. However, the matrices of the DG method are generally more sparse than the matrices of the FE method (see, e.g. [38]). In some cases, this might be advantageous.

3.2.7 Shape functions

To complete the definition of the DG method used in this work, we have to introduce the shape functions which span the local finite element spaces $\mathcal{S}(K_i)$ and $\mathcal{U}(K_i)$ of the finite element spaces V_h and Σ_h, respectively, introduced in (3.7)-(3.12). As already mentioned, the local finite element spaces $\mathcal{S}(K_i)$ and $\mathcal{U}(K_i)$ are polynomial spaces of order at most p. By definition, these spaces satisfy condition (3.68) which ensures existence and uniqueness of the solution. However, this implies that the shape functions need to be polynomial functions of order p at most. Since the framework of the DG method is based on numerical fluxes, the shape functions spanning the finite element spaces V_h and Σ_h do not have to satisfy further conditions. In particular, they do not even have to satisfy continuity conditions at the element interface, in contrast to shape functions for the FE method. In the following, we present our choice of shape functions and show that these shape functions span polynomial spaces.

In this work, the high-order hierarchical shape functions introduced in [93] are used to span the local finite element spaces $\mathcal{S}(K_i)$ and $\mathcal{U}(K_i)$. In [93] these shape functions are denoted as *H^1-conforming* and *$H(div)$-conforming shape functions for hexahedral elements*. Since these shape functions were originally built as shape functions for the FE method, the name indicates the H^1 and $H(div)$-conforming properties of the shape functions within a standard FE framework. As we use the shape function within our DG method, these conforming properties are not satisfied. Although these properties are not required for the DG method, they play a role if the DG hybridization introduced in Section 3.3.5 is applied to the DG method. Therefore, we will briefly introduce these properties at the end of this section.

Next, we introduce both the *H^1-conforming* and *$H(div)$-conforming shape functions* for hexahedral elements. The shape functions which are referred to as *H^1-conforming shape functions* span the local finite element spaces $\mathcal{S}(K_i)$ of V_h for each element $K_i \in \Omega_h$. The local finite element space $\mathcal{S}(K_i)$ on $K_i \in \Omega_h$ can be written as follows:

$$\mathcal{S}(K_i) = \mathcal{S}^n \oplus \left(\bigoplus_{i=1}^{12} \mathcal{S}_{e_i}^{p_{e_i}} \right) \oplus \left(\bigoplus_{i=1}^{6} \mathcal{S}_{f_i}^{p_{f_i}} \right) \oplus \mathcal{S}_c^{p_c}, \tag{3.72}$$

where $\mathcal{S}^n := span(\phi_i^v : i = 1, ..., 8)$ denotes the span of the vertex-based functions, $\mathcal{S}_{e_i}^{p_{e_i}}$ the span of the edge-based functions associated to edge e_i, $\mathcal{S}_{f_i}^{p_{f_i}}$ the span of the face-based functions associated to face f_i and $\mathcal{S}_c^{p_c}$ the span of cell based functions [93]. Of course, this implies that the space $\mathcal{S}(K_i)$ contains vertex-based shape functions, edge-based shape functions, face-based shape functions and cell-based shape functions. These shape functions are associated to each node, edge and face of the hexahedral element K_i as well as to the whole element. The whole element is denoted as cell in the following. Furthermore, all shape functions spanning $\mathcal{S}(K_i)$ are linearly independent [93]. Furthermore, within a standard FE method they can be applied using arbitrary polynomial approximation orders for each shape function associated to the edges p_{e_i}, faces p_{f_i} or cells p_c (see [93]).

Moreover, in [93] it is shown that the shape functions of $\mathcal{S}(K_i)$ form a basis of the tensor-product polynomial space if uniform approximation orders $p = p_{e_i} = p_{f_i} = p_c$ are used. The tensor product polynomial space $Q^{p,p,p}$ is given by

$$Q^{p,p,p}(K_i) = \{q_1 \cdot q_2 \cdot q_3 | q_j \in \mathcal{P}^p(K_i), j = 1, 2, 3\}. \tag{3.73}$$

The $H(\text{div})$-*conforming shape functions* span the local finite element spaces $\mathcal{U}(K_i)$ of Σ_h for each element $K_i \in \Omega_h$. The local finite element space $\mathcal{U}(K_i)$ of Σ_h on $K_i \in \Omega_h$ is defined by

$$\mathcal{U}^p(K_i) := \left(\bigoplus_{i=1}^{6} \mathcal{U}_{f_i}^{p_{f_i}} \right) \oplus \mathcal{U}_c^{p_c}, \tag{3.74}$$

where $\mathcal{U}_{f_i}^{p_{f_i}}$ and $\mathcal{U}_c^{p_c}$ denote the span of the face- and cell-based shape functions, respectively. In contrast to $\mathcal{S}(K_i)$, the local finite element space $\mathcal{U}(K_i)$ contains only shape functions which are associated to each face and the cell. The shape functions are also linearly independent and can also be applied within a standard FE framework using arbitrary polynomial orders, p_{f_i} and p_c, on each face and cell. Moreover, the $H(\text{div})$-*conforming* shape functions span the space

$$Q^{p+1,p,p} \times Q^{p,p+1,p} \times Q^{p,p,p+1} = \begin{pmatrix} Q^{p+1,p,p} \\ Q^{p,p+1,p} \\ Q^{p,p,p+1} \end{pmatrix} \tag{3.75}$$

for uniform polynomial approximation orders $p = p_{f_i} = p_c$ [93]. Entering (3.73) and (3.75) into (3.68) shows that condition (3.68) is satisfied and, therefore, the existence and uniqueness of the solution guaranteed. We refer to [93] for more detailed information on the shape functions. The complete definition of the shape functions can also be found in Appendix 5.2.

Note that the shape functions introduced above are defined on a hexahedral reference element given as $\hat{K} := \mathcal{H}[0, 1]^3$. To obtain the shape functions which are defined on each element $K_i \in \Omega_h$, the reference element shape functions need to be mapped to each element $K_i \in \Omega_h$.

The affine map which is used to map the shape functions to each element K_i is defined as follows:
For every $K_i \in \Omega_h$ a map $F_{K_i} : \hat{K} \longrightarrow K_i$ exists with

$$F_{K_i}(\hat{\mathbf{x}}) = B_{K_i}\hat{\mathbf{x}} + b_{K_i}, \tag{3.76}$$

where B_{K_i} is a non-singular diagonal 3×3 matrix and b_{K_i} is a vector and $\hat{\mathbf{x}}$ satisfies $\hat{\mathbf{x}} \in \hat{K}$, [60]. The Jacobian matrix of the mapping F_{K_i} is given as

$$J_{K_i} := det(dF_{K_i}) \quad \text{with} \quad dF_{K_i} = (\frac{\partial (F_{K_i})_l}{\partial \hat{x}_m})_{1 \leq l,m \leq 3}. \tag{3.77}$$

Furthermore, the unit outward normal $\mathbf{n}$ to K_i is defined using the unit outward normal $\hat{\mathbf{n}}$ to $\hat{K}$ for $\hat{\mathbf{x}} \in \partial \hat{K}$ [60]:

$$\mathbf{n}(F_{K_i}(\hat{\mathbf{x}})) = \frac{dF_{K_i}^{-T}(\hat{\mathbf{n}})}{|dF_{K_i}^{-T}(\hat{\mathbf{n}})|}(\hat{\mathbf{x}}). \tag{3.78}$$

Using the affine map, a transformation of a scalar shape function $p \in \mathcal{U}(K_i)$ can be expressed as

$$p(F_{K_i}(\hat{\mathbf{x}})) = \hat{p}(\hat{\mathbf{x}}), \tag{3.79}$$

where $\hat{p} \in \mathcal{U}(\hat{K})$ is the corresponding shape function on the reference element $\hat{K}$ [60]. Similarly, a vector shape function $\mathbf{u} \in \mathcal{S}(K_i)$ is related to a corresponding vector shape function $\hat{\mathbf{u}}$ on $\hat{K}$ by

$$\mathbf{u}(F_{K_i}(\hat{\mathbf{x}})) = \frac{1}{det(dF_{K_i})} dF_{K_i}\hat{\mathbf{u}}(\hat{\mathbf{x}}). \tag{3.80}$$

Note that the map in (3.80) is divergence conserving. This implies that the properties are conserved by mapping the vector function from the reference element $\hat{K}$ to the physical element K_i [60]. The presented mapping of the shape function to the physical element K_i using the affine map has the advantage that many terms can be calculated a priori on the reference element which saves computational costs with respect to time.
Next, we discuss the H^1- and $H(\text{div})$-conforming properties of the shape function for the standard FE method since they play a major role in the DG hybridization method presented in Section 3.3.5.
The H^1- and $H(\text{div})$- conforming properties of the shape functions reflect the physical properties of the electric potential and the flux density solution. The *H^1-conforming shape functions* satisfy the continuity conditions at the element interfaces when used within a standard FE method. This holds because only the shape functions which are associated to a certain face, edge or vertex have values unequal to zero on this specific face, edge or vertex, respectively [93]. All other shape functions vanish. Similarly, the *$H(div)$-conforming shape functions* ensure normal flux continuity at interfaces for the standard FE method. In this case, only the shape functions associated to a particular face have non-zero normal components on this face and all other shape functions which are not associated to that face vanish. Therefore, the degrees of freedoms of the associated shape functions solely determine the normal flux at this interfaces. Note that the use of shape functions which reflect the physical properties of the field quantities is important.

For instance, for the eigenvalue problem, the use of shape functions which doesn't conform to the field properties may lead to spurious modes [93].

The DG hybridization method exploits the H^1- and $H(\text{div})$-conforming properties to reduce the overall high number of DG DOF. More precisely, it transforms some DG DOF to to FE-type DOF. These FE-type DOF have to satisfy the H^1 and $H(\text{div})$-conforming continuity properties at the element interface. As our choice of shape functions satisfy the H^1 and $H(\text{div})$-conforming properties, the transformation of DG DOF to FE DOF can be performed easily. For that reason, we decided to use the shape functions in [93] with the H^1 and $H(\text{div})$-conforming properties. We refer to Section 3.3.5 for more details on the DG hybridization.

3.3 Boundary Conformal Discontinuous Galerkin method

In this section we introduce the boundary conformal Discontinuous Galerkin (BCDG) method which is based on the DG framework presented in Section 3.2. The BCDG method comprises several approaches, namely the cut-cell discretization approach, the adaptive approximation order method, the cell merging method and the optional DG hybridization method. In Section 3.3.1, we give a brief introduction to the BCDG method and a general literature overview. Section 3.3.2 shows the mathematical framework of the cut-cell discretization approach. The cut-cell discretization approach is applied to elements at material boundaries or interfaces to accurately approximate curved materials. More precisely, the elements at curved material boundaries or interfaces are split at the interface or boundary into sub-elements, denoted as cut-cells, which are treated as independent elements. In Section 3.3.3 we introduce the underlying geometry representation on which the cut-cell discretization is based. Furthermore, the numerical integration scheme and the implementation with regard to the geometry representation is described in greater detail. In the presence of geometrically small cut-cells, it turns out that the cut-cell discretization destroys the condition number of the system. To prevent ill-conditioning two additional approaches - the adaptive approximation order method and the cell merging method - are introduced in Section 3.3.4. Either of these two approaches can be applied to the cut-cell discretization approach. Finally, in Section 3.3.5 a DG hybridization method is presented which can be applied to the cut-cell discretization approach to reduce the overall high number of DG DOF in normal elements.

3.3.1 Introduction and literature review

The BCDG method presented in this work is based on three-dimensional structured hexahedral meshes that are not fitted to the material geometry and do not change during the simulation. Such meshes provide for greater efficiency since certain tasks such as node numbering or the assembly of the matrices are much easier to perform on structured hexahedral meshes than on other types of meshes [60]. In order to obtain sufficiently accurate results at curved material boundaries, an additional approach is applied to elements which are intersected by a material boundary or interface. In the following, we will refer to this approach as *cut-cell discretization approach*. The general idea is illustrated in Figure 3.1. It shows the computational domain of a quarter part of a cylindrical capacitor. The computational domain is given by two dielectric material layers represented by the green and blue coloured material domains, respectively. An exemplary structured hexahedral mesh represented by the black lines covers the computational domain. The material domains subdivide several elements into smaller sub-elements. These sub-elements are referred to as cut-cells in the following. They are treated as independent elements with its own set of approximation functions. Note that this procedure is only applied to elements at the material boundary or interface of the computational domain. The cut-cells are illustrated in Figure 3.1 using the chequer pattern. All other hexahedral elements are treated as described in Section 3.2. We have decided to choose the DG method as the underlying framework of the *cut-cell discretization approach* for the following two reasons: First, a DG approach allows for high order approximation. Second, it enables a flexible construction of the approximation spaces in which our *cut-cell discretization approach* can be naturally embedded.

However, when geometrically small cut-cells appear in the simulation, the *cut-cell discretization approach* faces the problem of ill-conditioning. The appearance of small cut-cells is a com-

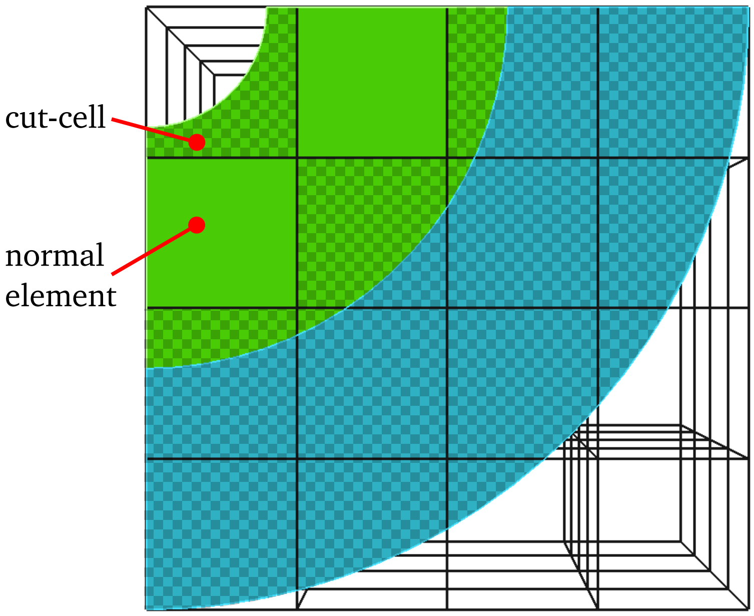

Figure 3.1.: Computational domain of the cylindrical capacitor verification example.

mon phenomenon in our approach, since the mesh is not fitted to the material and, therefore, material boundaries or interfaces intersect the element arbitrarily. To prevent ill-conditioning, we introduce two additional approaches, the *adaptive approximation order method* and the *cell merging method*, which are supplementary to the cut-cell discretization. The first approach is based on an adaptive approximation order scheme. This scheme lowers the approximation order in geometrically small cut-cells. The assigned approximation order in geometrically small cut-cells depends on the volume of the cut-cell. Specific thresholds for the volume of cut-cells are introduced which determine the approximation order in these cells. This approach turns out to be very simple from the implementation point of view and efficient in terms of condition number reduction The second approach - the *cell merging method* - merges geometrically small cut-cells with neighbouring elements. A threshold is introduced which specifies geometrically small cut-cells. For each small cut-cell, the neighbouring cell with the largest shared face is determined and merged to the geometrically small cut-cell. The merged cut-cell is considered as an independent cut-cell with its own set of approximation functions. The approximation order in the merged cut-cell remains the same. Note that either the *adaptive approximation order method* or the *cell merging method* are applied to the *cut-cell discretization approach*. Finally, a *DG hybridization method* is presented that couples the DG method with the finite element method (FEM). This is performed by an element-wise transformation of the DG degrees of freedom to FEM degrees of freedom in normal elements with local topological operators. Using the boundary conformal DG method with the additional hybridization scheme reduces the number of degrees of freedom of our system substantially.

The idea of *cut-cell discretization approach* builds on methods such as the partially filled cell (PFC) approach for FIT in [80], the locally conformal FDTD algorithm in [23] or the Finite Volume approach using cut cells in [66]. To approximate curved material boundaries on Cartesian grids, these methods also use additional approaches at the vicinity of the material boundaries or interfaces within their numerical scheme. In addition, in recent years several similar approaches for complex-shaped geometries have been developed. In the following, we discuss related approaches which are based on either the DG method or the Nitsche's method [62].

The DG-type methods closest to our BCDG method are the following: In [29] a cut-cell approach is introduced based on the DG method for the two-dimensional compressible Navier-Stokes equations on triangular meshes. The method uses meshes that do not conform to the underlying geometry boundary and, therefore, are cut at the boundary. The cut-cells are naturally embedded in the DG method and the boundary conditions are imposed weakly. A very similar approach on elliptic problems is proposed in [9] using a Baumann-Oden flux formulation on unfitted meshes.

Moreover, there have been a number of related contributions using the Nitsche's method. The Nitsche's method has first been introduced in [62] in 1971. Since then, it has been further developed to a general scheme for elliptic and parabolic equations [7] (see, e.g. [6]). The Nitsche's method is mentioned in this context, since it is similar to the DG method in many respects. For instance, it is also based on a variational or weak formulation and the approximation functions are allowed to be discontinuous across the element interfaces. Furthermore, some particular Nitsche-type methods can be formally regarded as special cases of the DG method (see, e.g. [48]). For the Nitsche's method several similar approaches using cut-elements to approximate complex-shaped geometries have been introduced. Note that these approaches are often referred to as fictitious domain finite element methods. For instance, [16] presents such a method for the two-dimensional Poisson equation using first-order approximations, [35] for the stationary heat conduction problem in two-dimensions for first-order approximations, [48] for the two- and three-dimensional Poisson equation for high order approximations and [57] for the Stokes problem in two and three dimensions for first-order approximations. Moreover, a similar approach based on an extended finite element method combined with a Nitsche's technique for ellitic equations in two dimensions has been proposed in [95]. In addition, several closely related methods for composite grid applications based on the Nitsche's method have been developed. These methods connect composite grids by an artificial interface which is specified by the edges of one mesh. The other mesh is then cut at the mesh interface, resulting in the emergence of cut-elements. Examples include the Nitsche-type methods proposed in [36] for the Poisson problem in two and free dimensions and in [57] for the Stokes problem in two or three dimensions. However, both methods use first-order approximations. In the field of computational electromagnetics, the approach closest to ours is a finite element method approach to the Finite-Difference Time-Domain scheme with cut-cells for the Maxwell's equations in two dimensions. This approach was published in [45] for first-order approximations. In this approach, the continuity at the element interfaces is enforced in the weak sense using the Nitsche's method.

The BCDG method we present in this section is similar to the above mentioned methods in the sense that we also consider elements which are cut at the material boundary or interface. However, having a closer look at the methods reveals that the methods differ from each other in many respects. Most obviously, they differ with respect to the equations they are applied to, the numerical schemes, the geometry representation and the numerical integration methods. The BCDG method we present is applied to the electro-quasistatic approximation of the Maxwell's equations. It is developed for a three-dimensional setting, whereas most other methods are presented for one or two-dimensional examples. It can also be applied to the electrostatic Maxwell's equations which are elliptic equations. Furthermore, we use a DG method with the LDG flux formulation and higher order approximation functions up to order three. Also the geometry representation and the numerical integration in our BCDG method differ from the methods introduced in the above mentioned papers. In [57] a numerical integration technique is presented which is slightly similar to the numerical integration scheme presented below. However, the numerical integration technique in [57] is so far restricted to first-order approximations, whereas in our case higher order approximations are used.

As mentioned previously, our *cut-cell discretization approach* faces the problem of ill conditioning when geometrically small cut-cells appear. This has also been reported for most of the above mentioned methods. Very recently, a few approaches have been introduced to prevent ill-conditioning. For instance, the fictitious domain method based on the Nitsche's method which introduced in [16] for two-dimensional elliptic interface problems uses additional ghost penalty terms. These ghost penalty terms are added to the gradient jumps in the boundary zone and yield a upper limit for the condition numbers [16]. A similar approach based on the Nitsche's method is presented in [57] for the Stokes problem in two and three dimensions. This approach also applies ghost penalties to gradient jumps. However, both methods are restricted to first order approximations. Furthermore, the ghost penalty terms have to be chosen carefully, since too high values have a negative effects on the accuracy and a too small values result in high condition numbers [16]. Another approach based on a combination of the XFEM method and the Nitsche's method for elliptic problems is presented in [95]. In this work, a suitable choice of penalty parameters and an additional diagonal preconditioner sufficiently stabilize the scheme. This method is also restricted to one- and two dimensional problems.

In September 2012, a very similar approach to our *cell merging method* has been presented in [48]. In this work, a method is introduced which is based on a so called Nitsche dG method for elliptic problems on structured quadrilateral or hexahedral meshes. Since geometrically small cut-elements are merged with neighbouring elements, this approach is very similar to the *cell merging method*. Furthermore, the authors prove an upper bound on the condition numbers using the additional merging technique. The results reported in this paper essentially correspond to the numerical results we obtain using the *cell merging method*. The evaluated condition numbers obtained from the BCDG method using the *cell merging method* are presented in Section 4.1.5. However, note that at the time when [48] was published, our *cell merging method* had already been developed and a paper describing this method had been accepted for publication (see, e.g. [32]).

Finally, we would like to present methods which are related to the optional *DG hybridization method* of the BCDG method. Of course, several methods coupling the DG method with the FE method exist (see, e.g. [28], [19]). The method closest to our method is proposed in [65]. In this work, a finite element formulation for the Poisson equation is presented which couples the LDG method with a standard conforming finite element method. In contrast to our optional *DG hybridization*, the domains to which the LDG method and the conforming FE method are applied to have to be specified a priori. The weak formulation and the coupling of the weak formulations is then built in accordance with the specified FE and LDG domains. Numerical examples in [65] show that, although the number of DOF is much smaller for the coupled method, the error of the coupled method is of the same order of magnitude as the error for the standard LDG method.

Note that this section is concerned with the presentation of the BCDG method which is applied to EQS and ES field problems with non-moving materials. However, the BCDG method presented in this section is also a key ingredient of the extension of the BCDG (EBCDG) method developed for field problems with materials that move over time. The EBCDG method is introduced in Section 3.6.

The advantages of the BCDG method are most obvious if the BCDG method is applied to field problems with moving materials. This is because most approaches which are applied to field problems with moving materials use standard FE methods on curved boundary conforming meshes where either the mesh is adapted to the moving material interface or a remesh is performed at each instant of time (see, e.g. [75]). By using the EBCDG method the usually costly mesh adaption and remeshing steps can be avoided. Up to our knowledge, the EBCDG method is the first contribution in the field of electromagnetic which presents a cut-cell approach for moving material distributions.

Nevertheless, the BCDG method applied to field problems with non-moving material distributions may, in some cases, also be superior to, for instance, FE methods on curved conforming meshes. Some real-world examples require the meshing of very complex shaped materials such as organs or blood vessels for biomedical applications. For such problems the BCDG method might benefit from that fact that no fitting of meshes to the material geometry is needed.

3.3.2 Cut-cell discretization approach

This section introduces the mathematical framework of the *cut-cell discretization approach*. Let Ω be a computational domain which contains N different material domains $\Omega_i, i = 1, ..., N$. The boundary of the computational domain is denoted as $\Gamma = \Gamma_D \cup \Gamma_N$ where Γ_D and Γ_N represent the boundary parts with Dirichlet and Neumann boundary conditions, respectively. In addition, the interfaces of the materials within the computational domain are represented by Γ_{Int}. Since a non-matching grid is used, some of the hexahedral elements of Ω_h contain parts of the interior material interfaces Γ_{Int} or of the boundary Γ. These interfaces or boundary parts of an exemplary element K_i are denoted as Γ_{K_i} to indicate the affiliation to element K_i and are defined as $\Gamma_{K_i} = K_i \cap (\Gamma \cup \Gamma_{Int})$.

To consider curved material boundaries or interfaces, we partition hexahedral elements which contain a part of the interface or boundary into one or more sub-elements. In the following these sub-elements are denoted as cut-cells. More precisely, each element K_i containing a part of the interface or boundary part Γ_{K_i} is split into one or more cut-cells K_i^C in the following way: First, the cut-cell domains K_i^C completely cover the hexahedral element domain:

$$K_i \cap \Omega_h = \cup_{i=1}^{N_{K_i}} K_i^C, \tag{3.81}$$

where N_{K_i} is the number of cut-cells resulting form the partition of element K_i. This number corresponds to the number of different material domains located in element K_i. Second, the cut-cells are always non-overlapping, i.e. the intersection of cut-cells satisfies

$$\Gamma_{K_i} = \cap_{i=1}^{N_{K_i}} K_i^C. \tag{3.82}$$

Figure 3.2 illustrates a possible partitioning of an exemplary hexahedral element K_i. The left-hand side of Figure 3.2 shows a computational domain of a water droplet Ω_W (blue material domain) on a insulation layer plate Ω_I (green material domain) surrounded by air Ω_A. The red shaded hexahedral element represents the exemplary element K_i which is split into cut-cells. The right-hand side of Figure 3.2 illustrates the cut-cell domains K_1^C and K_2^C in greater detail.

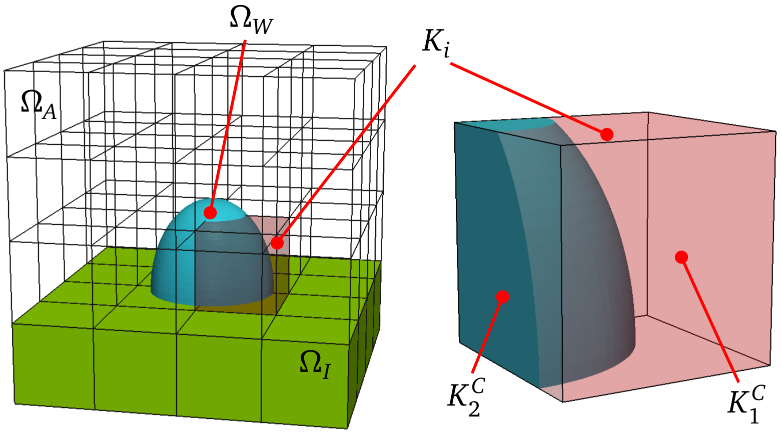

Figure 3.2.: Computational domain of a water droplet on a insulation layer plate surrounded by air Ω_A. The red shaded element presents an exemplary cut-cell domain.

Each cut-cell K_i^C is considered as an independent element with its own local finite element space and, therefore, it's own set of degrees of freedom. The element K_i is replaced by the cut-cells and is no longer considered. The new computational mesh including the cut-cells is given by

$$\Omega_h^C = \{K_i \in \Omega_h : K_i \cap \Omega_1 \neq \emptyset\} \cup \ldots \cup \{K_i \in \Omega_h : K_i \cap \Omega_N \neq \emptyset\}. \tag{3.83}$$

The set of normal elements Ω_h^N is the set of elements that are not intersected by a material interface

$$\Omega_h^N = \{K_i \in \Omega_h : \Gamma_{K_i} = \emptyset, K_i \cap \Omega_1 \neq \emptyset\} \cup \ldots \cup \{K_i \in \Omega_h : \Gamma_{K_i} = \emptyset, K_i \cap \Omega_N \neq \emptyset\}. \tag{3.84}$$

Having defined the set of elements of the new computational mesh including the cut-cells Ω_h^C and the set of normal elements Ω_h^N, the set of cut-cells can be written as

$$\Omega_h^{CC} = \Omega_h^C \setminus \Omega_h^N. \tag{3.85}$$

Of course, by including the additional cut-cell elements the number of elements in our new computational mesh Ω_h^C increases. The new number of elements in Ω_h^C is denoted by N_C and satisfies

$$N_C = N_N + N_{CC}, \tag{3.86}$$

where N_N is the number of normal elements and N_{CC} the number of cut-cells. Therefore, the new mesh can be written in the form:

$$\Omega_h^C = \{K_i\}_{i=1,\ldots,N_C}. \tag{3.87}$$

Moreover, the set of interior faces in Ω_h^C is given by

$$\begin{gathered}\mathcal{F}_0^C = \{f \cap \Omega_1 : f \in \mathcal{F}_0, f \cap \Omega_1 \neq \emptyset\} \cup \ldots \cup \{f \cap \Omega_N : f \in \mathcal{F}_0, f \cap \Omega_N \neq \emptyset\} \\ \cup \{K_i \cap \Gamma_{Int} : K_i \in \Omega_h, K_i \cap \Gamma_{Int} \neq \emptyset\}\end{gathered} \tag{3.88}$$

and the sets of boundary faces with Dirichlet and Neumann boundary conditions by

$$\mathcal{F}_D^C = \{K_i \cap \Gamma_D : K_i \in \Omega_h^C, K_i \cap \Gamma_D \neq \emptyset\}, \tag{3.89}$$
$$\mathcal{F}_N^C = \{K_i \cap \Gamma_N : K_i \in \Omega_h^C, K_i \cap \Gamma_N \neq \emptyset\}, \tag{3.90}$$

respectively. Finally, the set of faces associated with cut-cell elements is denoted by $\mathcal{F}^{CC}$.
Next, we introduce the approximate solutions ϕ_h and $\mathbf{D}_h$ on Ω_h^C. The approximate solutions ϕ_h and $\mathbf{D}_h$ are searched for in the finite element spaces

$$V_h = \{v \in L^2(\Omega) : v|_{K_i} \in \mathcal{S}(K_i)\ \forall K_i \in \Omega_h^C\}, \tag{3.91}$$
$$\Sigma_h = \{\boldsymbol{\tau} \in \left[L^2(\Omega)\right]^3 : \boldsymbol{\tau}|_{K_i} \in \mathcal{U}(K_i) \forall K_i \in \Omega_h^C\}, \tag{3.92}$$

where $\mathcal{S}(K_i)$ and $\mathcal{U}(K_i)$ represent the local finite element spaces. In normal elements, the local finite elements spaces are identical to the local finite element spaces of the DG method introduced in (3.7)-(3.8).
By contrast, the local finite element spaces $\mathcal{S}(K_i^C)$ and $\mathcal{U}(K_i^C)$ of a cut-cell $K_i^C \in \Omega_h^{CC}$ are defined differently. These local finite element spaces are spanned by the shape functions $v^C \in \mathcal{S}(K_i^C)$ and $\boldsymbol{\tau}^C \in \mathcal{U}(K_i^C)$, which are defined as

$$v^C = v \qquad \forall \mathbf{x} \in K_i^C, \tag{3.93}$$
$$v^C = 0 \qquad \forall \mathbf{x} \in K \setminus K_i^C, \tag{3.94}$$

and

$$\boldsymbol{\tau}^C = \boldsymbol{\tau} \qquad \forall \mathbf{x} \in K_i^C, \tag{3.95}$$
$$\boldsymbol{\tau}^C = \mathbf{0} \qquad \forall \mathbf{x} \in K \setminus K_i^C, \tag{3.96}$$

where K represents the hexahedral domain of the element which was split into cut-cells. In (3.93) and (3.95) the functions v and $\boldsymbol{\tau}$ represent shape functions which are defined on the hexahedron $\mathcal{H} := [0,1]^3$ and are mapped to the hexahedral element $K \in \Omega_h$ as shown in Section 3.2.7. This implies that inside the cut-cell domain the shape functions of cut-cell K_i^C are identical to standard shape functions defined on the hexahedral element $K \in \Omega_h$ with $K \notin \Omega_h^C$ containing the cut-cell K_i^C.
The approximate solutions of the BCDG method are given as the direct sum over all piecewise p-th order polynomial approximations $\phi_h^{K_i}(\mathbf{x}, t)$ and $\mathbf{D}_h^{K_i}(\mathbf{x}, t)$ as introduced in (3.9)-(3.10):

$$\phi_h(\mathbf{x}, t) = \oplus_{i=1}^{N_C} \phi_h^{K_i}(\mathbf{x}, t), \tag{3.97}$$
$$\mathbf{D}_h(\mathbf{x}, t) = \oplus_{i=1}^{N_C} \mathbf{D}_h^{K_i}(\mathbf{x}, t). \tag{3.98}$$

where for each cell $K_i \in \Omega_h^C$ the piecewise polynomial approximations $\phi_h^{K_i}(\mathbf{x}, t)$ and $\mathbf{D}_h^{K_i}(\mathbf{x}, t)$ can be expressed in terms of basis functions:

$$\phi_h^{K_i}(\mathbf{x}, t) = \sum_{i=1}^{N} \phi_h^i(t) v_i(\mathbf{x}), \tag{3.99}$$
$$\mathbf{D}_h^{K_i}(\mathbf{x}, t) = \sum_{i=1}^{N} \mathbf{D}_h^i(t) \boldsymbol{\tau}_i(\mathbf{x}). \tag{3.100}$$

Next, we present the semi-discrete weak formulation, the numerical flux formulation, the bilinear form and the matrix formulation for the *cut-cell discretization approach*. The semi-discrete weak DG formulation for the cut-cell discretization is obtained similarly as the semi-discrete weak DG formulation of the DG method (see Section 3.2.2). However, as we will see below, the weak formulation differs with respect to the test and approximation functions which belong to the finite element spaces V_h^C and Σ_h^C.

The weak DG flux formulation for the boundary conformal approach is defined as follows: Find $\phi_h(\mathbf{x},t) \in V_h$ and $\mathbf{D}_h(\mathbf{x},t) \in \Sigma_h^C$ such that $\forall K_i \in \Omega_h^C$

$$\int_K \boldsymbol{\tau} \cdot \frac{1}{\epsilon} \mathbf{D}_h dV = \int_K \nabla \cdot \boldsymbol{\tau} \phi_h d\mathbf{x} - \int_{\partial K} \tilde{\phi}_h \boldsymbol{\tau} \cdot \mathbf{n} dA \qquad \forall \boldsymbol{\tau} \in \Sigma_h^C, \quad (3.101)$$

$$\int_K \nabla \upsilon \cdot \left(\frac{\partial}{\partial t} \mathbf{D}_h + \frac{\kappa}{\epsilon} \mathbf{D}_h \right) dV = \int_{\partial K} \upsilon \left(\frac{\partial}{\partial t} \tilde{\mathbf{D}}_h + \frac{\kappa}{\epsilon} \tilde{\mathbf{D}}_h \right) \cdot \mathbf{n} dA \qquad \forall \upsilon \in V_h^C. \quad (3.102)$$

As already introduced in (3.19)-(3.20), the numerical fluxes $\tilde{\phi}_h$ and $\tilde{\mathbf{D}}_h$ $\forall K_i \in \Omega_h^C$ are defined by

$$\tilde{\phi}_h = \{\phi_h\} + \mathbf{C}_{12} \cdot [\![\phi_h]\!] \qquad \text{on } f_i \in \mathcal{F}_0^C, \quad (3.103)$$

$$\tilde{\mathbf{D}}_h = \{\mathbf{D}_h\} - \mathbf{C}_{12} [\![\mathbf{D}_h]\!] - C_{11} [\![\phi_h]\!] \qquad \text{on } f_i \in \mathcal{F}_0^C, \quad (3.104)$$

The flux formulation on boundary faces with Dirichlet and Neumann boundary conditions is identical to the flux formulation in (3.21)-(3.24) for all $f \in \mathcal{F}_D^C$ and for all $f \in \mathcal{F}_N^C$, respectively. The main difference to the standard DG method in normal elements is that the cut-cell approach incorporates, in addition to the usual numerical fluxes at normal cell interfaces, fluxes at the interior interface Γ_{Int}. Figure 3.3 illustrates the numerical fluxes, $\tilde{\phi}_h$ and $\tilde{\mathbf{D}}_h$, as vectors at the interior face Γ_{K_i} of the cut-cells K_1^C and K_2^C. Furthermore, cut-cells are only connected by numerical fluxes to elements with which they share a common face. Note that such a natural embedding of cut-cells is not feasible for the standard FE method, since the FE method requires certain continuity conditions at the element faces. The DG method, however, provides the flexibility for such a natural embedding of cut-cells.

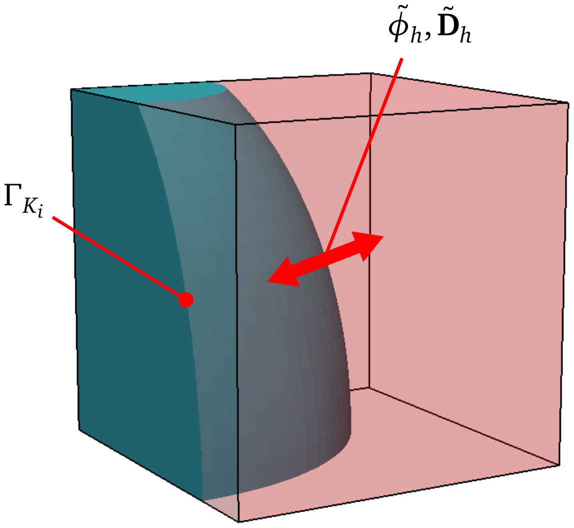

Figure 3.3.: Numerical fluxes at the interior face Γ_{K_i} of the cut-cells K_1^C and K_2^C.

Finally, the bilinear form of the BCDG method for the EQS approximation of the Maxwell's equations reads:
Find $\mathbf{D}_h(\mathbf{x}, t) \in \Sigma_h^C$, $\phi_h(\mathbf{x}, t) \in V_h^C$ such that

$$a\left(\frac{1}{\epsilon}\mathbf{D}_h, \boldsymbol{\tau}\right) = b\left(\phi_h, \boldsymbol{\tau}\right) + f\left(\boldsymbol{\tau}\right) \qquad \forall \boldsymbol{\tau} \in \Sigma_h^C, \tag{3.105}$$

$$-b\left(\upsilon, \left(\frac{\partial}{\partial t}\mathbf{D}_h + \frac{\kappa}{\epsilon}\mathbf{D}_h\right)\right) + s\left(\upsilon, \left(\frac{\partial}{\partial t}\phi_h + \frac{\kappa}{\epsilon}\phi_h\right)\right) = g(\upsilon) \qquad \forall \upsilon \in V_h^C, \tag{3.106}$$

where a, b, s, g and f are defined as introduced in (3.27)-(3.31), respectively. Note that the finite element spaces V_h^C and Σ_h^C differ from the finite elements spaces V_h and Σ_h used for the DG method. However, the weak DG formulation in matrix formulation for the boundary conformal DG method can be written as presented in (3.33)-(3.39):

$$\mathbf{M}\mathbf{I}_{1/\epsilon}\mathbf{d} = \mathbf{G}\boldsymbol{\phi} + \mathbf{f}, \tag{3.107}$$

$$-\left(\mathbf{G}^T\frac{d}{dt}\mathbf{d} + \mathbf{G}^T\mathbf{I}_{\kappa/\epsilon}\mathbf{d}\right) + \left(\mathbf{S}\frac{d}{dt}\boldsymbol{\phi} + \mathbf{S}\mathbf{I}_{\kappa/\epsilon}\boldsymbol{\phi}\right) = \mathbf{g}, \tag{3.108}$$

where $\mathbf{d}$ and $\boldsymbol{\phi}$ denote the potential and the flux density degrees of freedom, respectively. The terms $\mathbf{G}$, $\mathbf{M}$ and $\mathbf{S}$ represent the mass, gradient and stability term matrices which are defined as presented in (3.37)-(3.39) for all $\upsilon \in V_h^C$ and $\boldsymbol{\tau} \in \Sigma_h^C$.
Applying the Schur-complement to (3.107)-(3.108) yields the reduced semi-discrete weak DG formulation:

$$\left(-\mathbf{G}^T\mathbf{I}_\epsilon\mathbf{M}^{-1}\mathbf{G} + \mathbf{S}\right)\frac{d}{dt}\boldsymbol{\phi} = \left(\mathbf{G}^T\mathbf{I}_\kappa\mathbf{M}^{-1}\mathbf{G} - \mathbf{S}\mathbf{I}_{\kappa/\epsilon}\right)\boldsymbol{\phi} + \mathbf{g}_B. \tag{3.109}$$

Note that the bilinear form and the matrix formulation of the ES approximation can be derived in exactly the same way as the bilinear form and matrix formulation of the EQS approximation. Therefore, we only present these formulations without deriving them explicitly. The bilinear form of the BCDG method for the ES approximation satisfies:
Find $\mathbf{D}_h(\mathbf{x}, t) \in \Sigma_h^C$, $\phi_h(\mathbf{x}, t) \in V_h^C$ such that

$$a\left(\frac{1}{\epsilon}\mathbf{D}_h, \boldsymbol{\tau}\right) = b\left(\phi_h, \boldsymbol{\tau}\right) + f\left(\boldsymbol{\tau}\right) \qquad \forall \boldsymbol{\tau} \in \Sigma_h^C, \tag{3.110}$$

$$-b\left(\upsilon, \mathbf{D}_h\right) + s\left(\upsilon, \phi_h\right) = g(\upsilon) \qquad \forall \upsilon \in V_h^C, \tag{3.111}$$

where a, b, s, g and f are defined as introduced in (3.27)-(3.30) and (3.43), respectively. Furthermore, the discrete weak DG formulation for the ES approximation in matrix formulation using the finite element spaces V_h^C and Σ_h^C is given as follows:

$$\mathbf{M}\mathbf{I}_{1/\epsilon}\mathbf{d} = \mathbf{G}\boldsymbol{\phi} + \mathbf{f}, \tag{3.112}$$

$$-\mathbf{G}^T\mathbf{d} + \mathbf{S}\boldsymbol{\phi} = \mathbf{g}, \tag{3.113}$$

where the matrices $\mathbf{M}$, $\mathbf{G}$ and $\mathbf{S}$ and the vector $\mathbf{f}$ are defined as shown in (3.35)- (3.38) and (3.46), respectively.
Applying the Schur complement, reduces (3.112)-(3.113) to

$$\left(-\mathbf{G}^T\mathbf{I}_\epsilon\mathbf{M}^{-1}\mathbf{G} + \mathbf{S}\right)\boldsymbol{\phi} = \mathbf{g}_B. \tag{3.114}$$

where $\mathbf{g}_B$ is given by $\mathbf{g}_B = \mathbf{M}^{-1}\mathbf{I}_\epsilon\mathbf{f} + \mathbf{g}$.
When it comes to the evaluation of the integral terms in cut-cells presented above, a detailed specification of the cut-cell geometry and, moreover, a numerical integration scheme is needed.

3.3.3 Geometry representation and numerical integration techniques in cut-cells

In this section, the geometry representation and numerical integration techniques in cut-cells are introduced and discussed. We give an overview on established geometry representations in the field of computational engineering. Then, we present the geometry representation and the numerical integration techniques used in this work in greater detail.

3.3.3.1 Introduction and literature review

In most engineering fields CAD (Computer Aided Design) programs are used to model material geometry data. Modern CAD programs have become very successful, because they enable to draw very complex objects and assemblies easily and fast [43]. Furthermore, most CAD programs offer a huge variety of other geometric functions operating on the geometry models. It has become standard practice that designs are made with modern CAD programs and meshes are then built from these CAD design models [43]. Many of these CAD programs rely on explicit, parameterized geometry representations. Therefore, standard adaptive integration schemes can be used on these explicit geometry representations which ensure that the numerical integration error is small compared to the error of the conformal DG method.

As an alternative to CAD based geometry representations, a geometry representation which is based on the level set method [64] is also often used. The level set method has become popular in the field of computational fluid dynamics (CFD), especially for two-phase or multiphase flow applications. Some of the above mentioned papers on the Nitsche's method use such a level set method within their scheme (see, e.g. [16], [48]). The geometry is given by a smooth level set function ϕ. Within a specified region Ω, e.g. the liquid region, the level set function ϕ satisfies $\phi > 0$, whereas outside of this region it satisfies $\phi < 0$. The geometry interface $\Gamma = \partial\Omega$ is implicitly represented by the zero level set of the smooth function ϕ, i.e. $\Gamma(t) = \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\}$ [64]. The main advantage of the level set method is its simplicity. Furthermore, topology changes that typically occur if, for instance, water droplets merge or break, can be handled very easily [64]. On the other hand, the method is prone to larger numerical errors if the interface is stretched or torn heavily [79]. Nevertheless, accurate integration schemes for level set methods are still being developed. For instance, in [87] an efficient numerical integration scheme for piecewise linear interfaces has been introduced. In addition, numerical integration methods for curved interfaces exist. They are based on smoothed or discretized Heaviside functions and Dirac delta distributions (see, e.g. [81] and [26]). However, these methods may smear out the interface over a certain regularization zone [59]. This has a negative effect on the accuracy of the numerical integration scheme as it leads to approximated integrals which are not equivalent to the original integral at the interface [59]. In many applications examples these methods show to be first-order accurate in terms of computational costs (see, e.g. [26] and [59]).

In this work, we have decided to use the geometry representation of the open source CAD program Open CASCADE Technology (OCCT) [2] for the following reasons: First, the integration schemes of level set methods are still being developed and the integration schemes known to us do not guarantee more than first-order accuracy. Second, the level set method is mainly used in the field of computational fluid dynamics, whereas in the field of computational electromagnetics material models are usually available as CAD models. So far, there is no CFD solver available which uses a cut-cell discretization on the level set method for the water droplet example.

Therefore, we were not able to couple our code with a CFD solver using a similar cut-cell discretization based on the level set representation. Furthermore, no standard geometry libraries for level set representations exist which offer such a huge variety of geometric applications as standard CAD programs.

However, we have developed our code in such a way that it is completely open to any other kind of geometry representation and numerical integration scheme. Once a CFD code using a similar cut-cell approach is available, it can be easily implemented in our code in reasonable time. In addition, since the BCDG method uses a geometry representation based on CAD models, it can be applied to various other field problems in the field of computational electromagnetics and is, therefore, not restricted to the water droplet example.

3.3.3.2 Geometry representation of cut-cells

Next, we present the features of the CAD program Open CASCADE Technology (OCCT) [2] which is used for the geometry representation of cut-cells. In this section we closely follow the presentation of OCCT in [2]. OCCT is an open source C++ class library which provides components for two or three-dimensional geometric modelling, visualization, geometric analysis application and data exchange. The geometry objects of OCCT are complying with the STEP format specification. Furthermore, OCCT is open to standard format CAD design models since it includes import and export functions to and from standard formats such as IGES, STL and STEP.

The geometry representation used in this work is based on the boundary representation (B-Rep). It is composed of geometry and topology data structures. The geometry data of OCCT provides a great variety of parametrized 2D and 3D geometric objects which are naturally oriented and handled by reference. The provided geometric objects range from very simple curves and surfaces to Bezier, BSplines as well as offset curves and surfaces. The topology data structure specifies the relationship between the geometric objects and how the geometric objects are connected. The geometric models described by the topology data structure are denoted as shapes. These shapes can be further decomposed into the following topology components (see [2]):

- vertices: refers to points in geometry.
- edges: refers to curves which are bounded by vertices.
- wires: sequences of edges.
- faces: refers to surfaces which are bounded by wires.
- shells: assembly of faces which are linked together by edges of their wire boundary.
- solids: 3D geometry object which is defined by an enclosed shell.
- compound solids: collection of solids.

For instance, a solid is defined by an enclosed shell which is an assembly of faces. The faces are represented by surfaces bounded by wires which in turn refer to curves bounded by vertices.

Furthermore, OCCT provides an extensive selection of operations and functions which operate on B-Rep data structures (shapes). For instance, the operations and functions available in OCCT include *Boolean Operations*. These operations are used to model complex shapes using simpler given shapes. In this work, the *Common and Cut Boolean Operations* are used. They create a new shape from two existing shape by subtracting one shape from the other (*Cut Boolean Operation*) or adding two shapes together (*Common Boolean Operation*). The resulting shapes of the *Cut* and *Common Boolean Operation* are also B-Rep shapes.

Moreover, OCCT provides functions to compute the derivative vectors up to second degree as well as the normal vector for each parameter point on a curve or surface. These vectors can then be used in the numerical integration schemes to evaluate the integrals over the B-Rep shapes of cut-cells. For more information on the geometry kernel and the embedded geometric data structures we refer to [2].

3.3.3.3 Geometry implementation

In this section, we describe in detail how the geometry representation and the related geometric functions are embedded in the BCDG method. We begin with the read-in process at the beginning of a simulation. In our implementation the mesh and the material geometry are treated separately. The BCDG method uses the functions of OCCT to import CAD models of the materials which are available in the standard CAD model formats such as STEP, STL, IGES or B-Rep.

Then, OCCT is used to detect the material affiliation of each node in the mesh. This is necessary to find all hexahedral elements at the material boundary or interface. More precisely, the information about the node material affiliation is used to detect the hexahedral elements at the material interfaces $K_i \in \Omega_h \setminus \Omega_h^N$. This search is conducted in an element-wise manner. If an element contains different material parameters, it is considered as an element at the boundary that needs to be split into cut-cells.

The detection of the node material affiliation is performed using a technique which is often referred to as ray-shooting or ray-casting (see, e.g. [55]). More precisely, the curve-solid intersection functions of OCCT are applied for this procedure. Since structured Cartesian grids are used, the edge which is linked to a certain node in one particular coordinate axis direction can be used to define the orientation of a straight curve or ray. The length of this curve is chosen to be greater than the cross section of the computational domain so that the curve runs through the whole computational domain in one direction. Because of the structured layout of the nodes in the mesh, several nodes are located on this curve. After having defined the curve, the *curve-solid intersection* function is applied to the curve and the computational domain. Note that this *curve-solid intersection* function has to be performed only once for all nodes on this curve. Next, the intersection points of the curve with the existing material domains are evaluated and counted. As the curve is a straight line and oriented, the number of intersection points and the signed distance of the nodes to the intersection points jointly reveal whether a node is inside or outside of a specific material domain. If a node is inside of a specific material domain, it has an odd number of positive and negative distance values to the intersections points associated with the material domain. By contrast, if a node is outside of a specific material domain, it has an even number of positive and negative distance values to the intersections points associated with the material domain.

Next, we explain how hexahedral elements $K_i \in \Omega_h \setminus \Omega_h^N$ are split into cut-cells by the material interface or boundary interface. The splitting procedure is conducted using the *Boolean Operations* of OCCT. First, the node coordinates of an hexahedral element $K_i \in \Omega_h \setminus \Omega_h^N$ are used to model the hexahedral element and its faces as B-Rep shapes. Then the *Boolean Operations*[3] are applied to both the B-Rep shapes of the hexahedral element K_i and the B-Rep shapes of the material geometry. The B-Rep shapes obtained from this Boolean Operation represent the cut-cell domains.

Note that we extract and save the faces of cut-cell domains $f_i \in \mathcal{F}^{CC}$ instead of the B-Rep solids. Since B-Rep solids are represented as enclosed assemblies of faces, it is sufficient to save only the faces of cut-cell elements $f_i \in \mathcal{F}^{CC}$. From an implementation point of view, this proves to be more efficient since the assembly routine can be applied over all faces $f_i \in \mathcal{F}^C = \mathcal{F}_D^C \cup \mathcal{F}_0^C \cup \mathcal{F}_N^C$ instead of over all elements. Such a routine is commonly denoted as a face-wise assembly routine, whereas the assembly over all elements is called cell-wise routine. However, because a face is shared by two elements (unless it is a boundary face) the numerical integration scheme on a particular face is performed only once for the face-wise assembly routine. If the assembly routine had been applied in a cell-wise manner, the numerical integration on a face of a cut-cell were to be conducted at least twice. Therefore, by using an face-wise assembly routine, the computational cost are reduced in a natural way (for more details, see Section 3.3.3.4).

To distinguish between the original computational mesh Ω_h and the cut-cell mesh Ω_h^C, the information of the new mesh is treated internally as an additional mesh with its own independent numbering. Additional functions are introduced which connect the original computational mesh Ω_h with the cut-cell mesh Ω_h^C. The information of the cut-cell mesh Ω_h^C is restricted to the minimum required information to increase the efficiency of the approach.

3.3.3.4 Adaptive Gaussian quadrature on geometric cut-cell representation

In Section 3.3.2, we have introduced the bilinear form and the matrix formulation of the BCDG method in (3.25)-(3.31) and (3.33)-(3.40), respectively. These formulations require the evaluation of integrals over the cut-cell domains and the cut-cell faces. However, the cut-cell domains which result from the *Boolean Operations* are often arbitrarily and complex shaped. Since standard quadratures of the DG or FE method are tailor-made for the application to certain geometric element objects such as tetrahedra or hexahedra, they can not be applied to the cut-cells domains without major modifications. A widely used approach to approximate the integrals in (3.33)-(3.39) is to construct the cut-cell domain or faces by meshes of smaller and simpler elements that conform with the cut-cell boundary. For instance, in [48] and [9] triangular and tetrahedral meshes with curved boundaries are used to model the cut-cell. These meshes are fitted to the cut-cell boundary. The numerical integration of the integrals over the cut-cell domain and faces are then performed by using quadratures on the mesh elements covering the cut-cell. However, since each of the cut-cells has to be meshed separately, these approaches turn out to be computationally costly.

For this reason, we decided to implement an alternative method in our code. Since we use a geometry representation based on B-Rep shapes, the integrals over the cut-cell faces in (3.105)-

[3] Note that the *Common Boolean Operation* is applied to the material geometry of the material with the highest internal number. The other cut-cell could be extracted using the *Cut Boolean Operation*. However, this is not necessary, as we can calculate the integrals of the cut-cell with the lower material number by subtracting the integrals of the cut-cell with the higher material number from the integral of the hexahedral element.

(3.109) can be evaluated using an adaptive Gauss-Legendre quadrature which is provided by OCCT. The surfaces ∂K_i^C of a cut-cell $K_i^C \in \Omega_h^{CC}$ can be decomposed into faces f_i which satisfy

$$\partial K_i^C = \cup_i^n f_i(u_i, v_i), \tag{3.115}$$

where u_i and v_i denote the parametric direction of the face. Since the faces $f_i(u_i, v_i)$ are oriented and parametrized, the adaptive Gauss-Legendre quadrature can be directly applied to these faces. For instance, the flux integral term in (3.36) restricted to cut-cell $K_i^C \in \Omega_h^{CC}$ can be calculated by transforming the integral term to a integral term over the parametrization. This transformation is given by

$$\begin{aligned}\int_{\partial K_i^C} \left(\{v_j\} + \mathbf{C}_{12} \cdot [\![v_j]\!]\right) [\![\boldsymbol{\tau}_i]\!] ds &= \sum_i \int_{f_i} \left(\{v_j\} + \mathbf{C}_{12} \cdot [\![v_j]\!]\right) [\![\boldsymbol{\tau}_i]\!] ds \\ &= \sum_i \int_{u_i} \int_{v_i} \left(\{\hat{v}_j\} + \mathbf{C}_{12} \cdot [\![\hat{v}_j]\!]\right) [\![\hat{\boldsymbol{\tau}}_i]\!] \mathbf{J} du_i dv_i\end{aligned} \tag{3.116}$$

where $\mathbf{J}$ is the Jacobian matrix of the transformation and (u_i, v_i) the parameters of face f_i (see, e.g. Section 3.2.7 or [52]). As OCCT provides functions computing the normal and the derivative vectors on each parameter point (u_i, v_i) of the face f_i, the application of the adaptive Gaussian quadrature to the term in (3.110) is straightforward.

The adaptive Gauss-Legendre quadrature provided by OCCT internally decomposes each face f_i into smaller parametrized patches p_j satisfying $f_i = \cup_j p_j$. The decomposition into patches depends on the embedded geometric object data structure. Furthermore, for each of the patches p_j a function determines the number of Gauss points N_{GP} needed to achieve a good accuracy in each of its parametric directions u_j and v_j [2]. In addition, an a priori specified global tolerance value I_ϵ is defined to determine whether an approximation of (3.116) is acceptable or not. This process is conducted using the bisection procedure: First, the Gauss-Legendre quadrature is performed recursively on each patch with N_{GP} and with $2/3N_{GP}$ Gauss points. Then, the relative difference dI of the two approximations I_1 and I_2 of (3.110) on N_{GP} and $2/3N_{GP}$ Gauss points is evaluated. The relative difference dI is given by

$$dI = abs(I_1 - I_2)/abs(I_1). \tag{3.117}$$

If the relative difference dI is smaller than the specified tolerance I_ϵ, then the more accurate approximation I_1 of the patch integral is accepted. If this is not the case, the patch is further decomposed. Let $[u_{j1}, u_{j2}]$ be the parameter interval of patch p_j not satisfying the tolerance criterion. The parameter interval is then divided into two parts, $[u_{j1}, c]$ and $[c, u_{j2}]$, with $c = (u_{j1} + u_{j2})/2$. This decomposition procedure can only be performed until a certain minimum patch size is reached. Hence, the algorithm terminates with an approximation of the integral in (3.116) if either the relative difference dI on all patches p_j is smaller than the specified tolerance value [2] or if the minimum patch size is reached.

However, the integral terms over the cut-cell volumes in (3.35)-(3.37) require a separate treatment. As the geometry kernel [2] doesn't provide a parametrized volume representation of the cut-cells, the volume integrals are evaluated over the parametrized boundary representation. Therefore, the cut-cell volume integrals in (3.35)-(3.37) are transformed using the divergence theorem (see, e.g. [52]). This method is widely used in the field of computer graphics to evaluate integral properties of solids which are given as B-Rep data structures [52]. These integral properties include the volume and the moment of inertia of B-Rep data structures. The divergence theorem states that for any given compact bounded three-dimensional submanifold the following equation holds (see, e.g. [47]):

$$\int_{K_i^C} \nabla \cdot \mathbf{F} dV = \oint_{\partial K_i^C} \mathbf{F} \cdot \mathbf{n} ds = \sum_i \int_{f_i} \mathbf{F} \cdot \mathbf{n}_i ds, \tag{3.118}$$

where $\mathbf{F}$ is a well-defined vector field on K_i^C and $\mathbf{n}$ the unit outward normal of ∂K_i^C. Note that in our case the cut-cell volumes $K_i^C \in \Omega_h^C$ which are enclosed by its boundary faces ∂K_i^C are the compact bounded three-dimensional submanifolds. In the following, we assume that we are given hexahedral meshes and material geometries which ensure that the divergence theorem can be applied. If this is not the case, the hexahedral mesh or the material geometry has to be changed to satisfy the necessary conditions.

Next, the transformations of the matrix formulation in (3.33)-(3.39) are presented. Using the divergence theorem, the matrix components of $\mathbf{M}$ and $\mathbf{G}$ corresponding to cut-cell K_i^C are transformed to the following form:

$$\begin{aligned}
\mathbf{M}_{i,j|K_i^C} &= \int_{K_i^C} \tau_i \cdot \tau_j dx = \int_{\partial K_i^C} \mathbf{F}_{i,j} \cdot \mathbf{n} dx, & (3.119)\\
\mathbf{G}_{i,j|K_i^C} &= \int_{K_i^C} \nabla \cdot \tau_i \upsilon_j d\mathbf{x} - \int_{\partial K_i^C} \left(\{\upsilon_j\} + \mathbf{C}_{12} \cdot [\![\upsilon_j]\!]\right) [\![\tau_i]\!] dA & (3.120)\\
&= \int_{\partial K_i^C} \left(\mathbf{P}_{i,j} \cdot \mathbf{n} + \left(\{\upsilon_j\} + \mathbf{C}_{12} \cdot [\![\upsilon_j]\!]\right) [\![\tau_i]\!]\right) dA, & (3.121)
\end{aligned}$$

where $\mathbf{F_{i,j}}$ and $\mathbf{P}_{i,j}$ are vector functions satisfying

$$\begin{aligned}
\nabla \cdot \mathbf{F}_{i,j} &= \tau_i \cdot \tau_j, & (3.122)\\
\nabla \cdot \mathbf{P}_{i,j} &= \nabla \cdot \tau_i \upsilon_j, & (3.123)
\end{aligned}$$

Since the shape functions are polynomials spanning the polynomial finite element spaces $\mathcal{S}(K_i^C)$ and $\mathcal{U}(K_i^C)$, it is always possible to determine vector functions $\mathbf{F}_{i,j}$ and $\mathbf{P}_{i,j}$. However, they are not uniquely defined. Using the vector functions $\mathbf{F}_{i,j}$ and $\mathbf{P}_{i,j}$, the semi-discrete weak DG formulation in (3.105)-(3.109) corresponding to cut-cells can be fully transformed to surface integrals over the cut-cell faces. These surface integrals are then evaluated using the adaptive Gauss-Legendre quadrature as presented above.

Note that the adaptive Gauss-Legendre quadrature is performed adaptively once on each cut-cell face f_i for a single component of one matrix in (3.35)-(3.39). When a sufficient accurate approximation of this matrix component is found, the Gauss points are saved and used to evaluate the remaining components of the matrices in (3.35)-(3.39) in a standard Gauss-Legendre quadrature fashion. This procedure saves computational costs and leads to equivalent results with respect to accuracy.

From an implementation point of view, when calculating the product of the shape functions in Section 3.2.7, i.e. $\tau_i(\mathbf{x}) \cdot \tau_j(\mathbf{x})$ with $\mathbf{x} = (x_1, x_2, x_3)^T \in K_i^C$, we are able to write this product as

$\tau_i \cdot \tau_j = f(x_1)g(x_2)l(x_3)$, where f, g and l are polynomial functions depending only on x_1, x_2 and x_3, respectively. In this case the primitive functions $S_f(x)$, $S_g(x_2)$ and $S_l(x_3)$ of f, g and l, respectively, can be easily determined. Using these primitive functions, a vector function $\mathbf{F}_{i,j}$ satisfying (3.122)-(3.123) can be written as

$$\mathbf{F}_{i,j} = \frac{1}{3}\begin{pmatrix} S_f(x_1)g(x_2)l(x_3) \\ f(x_1)S_g(x_2)l(x_3) \\ f(x_1)g(x_2)S_l(x_3) \end{pmatrix}. \tag{3.124}$$

Note the vector functions $\mathbf{F}_{i,j}$ and $\mathbf{P}_{i,j}$ are calculated only once at the beginning of a simulation. The resulting vector functions are then saved. To calculate these vector functions fast and efficiently, the implementation of the shape functions in [68] is used (see, e.g. [93] or Section 3.2.7).

3.3.4 Numerical treatment of geometrically small cut-cells

Since the *cut-cell discretization approach* divides hexahedral elements at the boundary or interface of arbitrarily shaped materials, geometrically small cut-cells may arise within the computational mesh Ω_h^C. As mentioned above, the existence of geometrically small material domains deteriorates the problem condition and may lead to simulation breakdowns or numerical instabilities. This comes as no surprise since the support of the basis functions in cut-cells can be arbitrarily small. This issue of ill-conditioning does not concern our *cut-cell discretization approach* only. In fact, all of the above discussed DG or Nitsche's methods which use a discretization where elements are cut at the interface or boundary face the problem of ill-conditioning (see, e.g the immersed or fictitious domain methods in [48], [57] or the finite cell method in [25]). In Section 3.3.1, we have presented several approaches of other authors which prevent ill-conditioning. The reader is referred to Section 3.3.1 for more details on that issue.

In the following we present two additional approaches for the *cut-cell discretization approach* to overcome ill-conditioning in the presence of small cut-cells. We start with the *adaptive approximation order method* which is based on an adaptivity scheme. Afterwards the *cell merging approach* is presented which is based on a merging technique. Both additional approaches are supplementary to the *cut-cell discretization approach*. We would like to point out that the additional approaches cannot be applied together. Either the adaptive approximation order or the cell merging approach are used to reduce the condition number. Numerical results of both approaches can be found in Section 4.1. We also refer to Section 4.1 for a detailed discussion on the numerical results and the condition numbers.

To track geometrically small cut-cells, we use a function provided by OCCT [2] which evaluates the approximate volume of B-Rep data structures (see, e.g. Section 3.3.3 or [2], for further information on the geometry representation). This function is called at the beginning of a simulation, just after the classification of elements into cut-cells or normal elements.

3.3.4.1 Adaptive Approximation Order method

The *adaptive approximation order method* has certain similarities to well-known adaptive strategies such as the *h*, *p* and the *hp*-adaptivity where on the basis of a given error estimate either the size of the grid (*h-adaptivity*), the polynomial order (*p-adaptivity*) or both the size of the grid and the polynomial order (*hp-adaptivity*) are adapted to decrease the overall error of the computation. An overview on that issue and recent error estimation schemes for the DG method can be found in [38]. Our approach is less related to the *h-adaptivity*. Obviously, we are not interested in reducing the computation grid size since we operate on a fixed grid. Our approach is rather related to the *p-adaptivity*. However, in contrast to standard p-adaptive strategies, we do not use an error estimate. In our case a lower approximation order is assigned to the cut-cells which are considered as geometrically small compared with the normal element size. The overall approximation order in all other cells is kept fixed.

More precisely, the classification of geometrically small cut-cells is determined using thresholds $\mathcal{T}_i \in \mathbb{R}, i = 0,..,N_{pn} = p_N - 1$ for the volume of cut-cells. A particular threshold $\mathcal{T}_i$ is used for each possible approximation order p_i smaller than the overall normal approximation order p_N. The thresholds $\mathcal{T}_i$ are determined using the volume information of cut-cells which is provided by OCCT [2] (see Section 3.3.4). This choice is quite efficient, as the volume is anyway calculated to check whether an element contains an interface or is just touching an interface. Since the threshold depends on the shape functions and their approximation order, it is difficult to determine general threshold values. We decided to use threshold values which are independent of the hexahedral grid size. The thresholds are given as $\mathcal{T}_i = t_i V_n$ where V_n denotes the volume of a normal hexahedral element and $t_i \in \mathbb{R}$. In our case, the threshold values t_i are determined by performing calibration tests on a verification example (see Section 4.1). Of course, other threshold values or other ways to determine the threshold values would be feasible.

Given the threshold values $\mathcal{T}_i$, the overall set of cut-cells which are considered as small is given by

$$\Omega_h^{CCSmall} = \Omega_h^{CCp_0} \cup \Omega_h^{CCp_1} \cup ... \cup \Omega_h^{CCp_{N_{pn}}}, \tag{3.125}$$

where the sets of small cut-cells for the individual threshold values are given as

$$\Omega_h^{CCp_0} \quad = \quad \{K_i^C \in \Omega_h^{CC} : \int_{K_i^C} 1 dV \ < \mathcal{T}_0\}, \tag{3.126}$$

$$\Omega_h^{CCp_i} \quad = \quad \{K_i^C \in \Omega_h^{CC} : \mathcal{T}_{i-1} < \int_{K_i^C} 1 d\mathbf{x} \ < \mathcal{T}_i\} \qquad \forall i = 1,...,N_{pn}. \tag{3.127}$$

As a result, the set of cut-cells is split into $N_{pn} + 1$ sets of small cut-cells and into a set of normal cut-cells denoted as Ω_h^{CCN} satisfying

$$\Omega_h^{CCN} = \Omega_h^{CC} \setminus \Omega_h^{CCSmall}. \tag{3.128}$$

We assign shape functions of approximation order $p_i < p_N$ for $i = 1,...,N-1$ to the set of small cut-cells $\Omega_h^{CCp_i}$. Furthermore, the general staircasing approximation is applied to the set of cut-cells $K_i^C \in \Omega_h^{CCp_0}$. Note that the threshold $\mathcal{T}_0$ is set to a very small value close to zero. Therefore, the influence of the staircasing approximation in these cut-cells is negligible.

The *adaptive approximation order method* clearly exploits the flexibility of the DG scheme. Since no continuity conditions are needed, the approximation order can be lowered in any element without any restrictions. We would like to point out that the *adaptive approximation order* *method* can also be applied in combination with the *DG hybridization method* which is introduced in Section 3.3.5. The *DG hybridization method* reduces DG degrees of freedom to FE degrees of freedom in elements away from the interface or boundary. Since we use hierarchical high order shape functions, which can be applied using arbitrary polynomial orders on each edge, face and cell within a standard FE framework [93], the *adaptive approximation order method* doesn't cause any problems in such a setting.

Furthermore, note that the *adaptive approximation order method* only lowers the approximation order p in small cut-cells $K_i^C \in \Omega_h^{CCSmall}$. Since the shape functions span the local finite element spaces, only the local finite element spaces in these small cells change. Therefore, the bilinear form and the matrix formulation of the boundary conformal DG method with the additional adaptive scheme are defined as presented in Section 3.3.2 and 3.2.4. The system of equations is solved as shown in Section 3.2.4. This also implies that the implementation of the *adaptive approximation order method* is straightforward and doesn't impose any significant additional computational costs.

3.3.4.2 Cell Merging method

In this section, we present the *cell merging method* which is based on merging geometrically small cut-cells with larger neighbouring elements. The numerical framework of this method is presented in the following.

As for the *adaptive approximation order method*, geometrically small cut-cells need to be determined. However, in this case only a single threshold is used to classify geometrically small cut-cells. This threshold is denoted by $\mathfrak{T}_M \in \mathbb{R}$ and is given as

$$\mathfrak{T}_M = t_M V_n, \tag{3.129}$$

where V_n denotes the volume of a normal hexahedral element. The threshold value $t_M \in \mathbb{R}$ is specified by performing a calibration on specific test examples as described above. Note that there are many other ways to classify geometrically small cut-cells or to determine threshold values. However, numerical examples using this approach show that the threshold based on the volume yields good results.

The set of geometrically small cut-cells is given by

$$\Omega_h^{CCSmall} = \{K_i^C \in \Omega_h^{CC} : \int_{K_i^C} 1 dV \ < \mathfrak{T}_M\}. \tag{3.130}$$

Having determined the set of geometrically small cut-cells, each of these cut-cells in the set is merged to a neighbouring element according to the following rule: Among the neighbouring elements having the same material, a geometrically small cut-cell $K_i^C \in \Omega_h^{CCSmall}$ is merged to the element the small cut-cell K_i^C shares its largest face with. This implies that the relevant neighbouring element K_i^N of K_i^C satisfies

$$K_i^N = \max_{K \in (\Omega_h^C \setminus \Omega_h^{CCSmall})} \int_{\partial(K \cap K_i^C)} 1 dV \tag{3.131}$$

with $\epsilon_{|K_i^N} = \epsilon_{|K_i^C}$ and $\kappa_{|K_i^N} = \kappa_{|K_i^C}$. Therefore, the set of neighbouring elements of small cut-cells K_i^C is defined by

$$\Omega_h^{CCNeigh} = \{K_i^N \in \Omega_h^C \setminus \Omega_h^{CCSmall} : K_i^N = \max_{K \in (\Omega_h^C \setminus \Omega_h^{CCSmall})} \int_{\partial(K \cap K_i^C)} 1 dV, \epsilon_{|K_i^N} = \epsilon_{|K_i^C}, \kappa_{|K_i^N} = \kappa_{|K_i^C}\}. \tag{3.132}$$

Figure 3.4 illustrates such an exemplary geometrically small cut-cell $K_i^C \in \Omega^{CCSmall}$ which is merged to element K_i^N. The two cut-cells, K_i^C and K_i^N, are then replaced by the merged cut-cell $K_i^M = K_i^C \cup K_i^N$, which is considered as an independent cut-cell with its own set of basis functions.

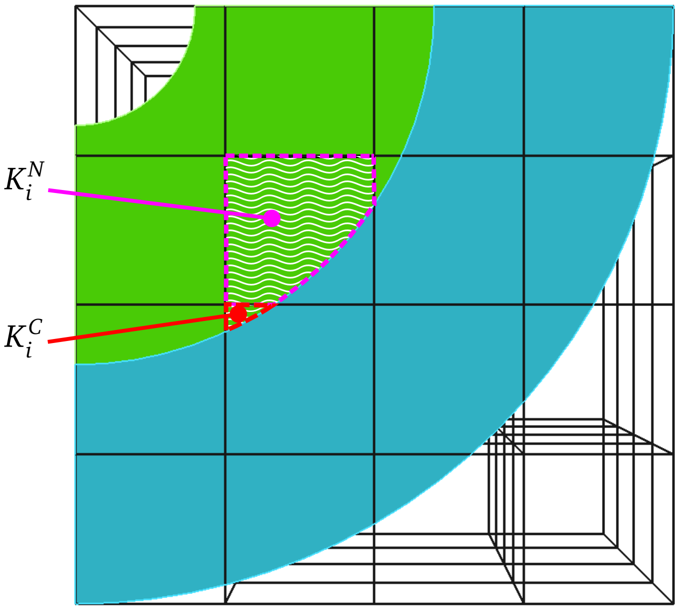

Figure 3.4.: Merging of an exemplary geometrically small cut-cell $K_i^C \in \Omega^{CCSmall}$ to the neighbouring element K_i^N.

The resulting computational mesh including the merged cells is given by

$$\Omega_h^{CM} = \mathcal{N} \cup \{K_i^M \in \Omega_h^C \setminus \mathcal{N} : K_i^M = K_i^C \cup K_i^N \ \forall K_i^C \in \Omega_h^{CCSmall},\ \forall K_i^N \in \Omega_h^{CCNeigh}\}, \tag{3.133}$$

where

$$\mathcal{N} = \Omega_h^C \setminus (\Omega_h^{CCSmall} \cup \Omega_h^{CCNeigh}). \tag{3.134}$$

is the set of elements which were not involved in the merging process.
Next, the *cut-cell discretization approach* as presented in Section 3.3.2 is applied to Ω_h^{CM}. The finite element spaces of the boundary conformal DG method with the additional *cell merging method* can be written as

$$V_h^{CM} = \{v \in L^2(\Omega) : v|_{K_i} \in \mathcal{S}(K_i)\ \forall K_i \in \Omega_h^{CM}\}, \tag{3.135}$$

$$\Sigma_h^{CM} = \{\tau \in \left[L^2(\Omega)\right]^3 : \tau|_{K_i} \in \mathcal{U}(K_i) \forall K_i \in \Omega_h^{CM}\}. \tag{3.136}$$

where $\mathcal{S}(K_i)$ and $\mathcal{U}(K_i)$ represent the local finite element spaces. The shape functions υ^M and $\boldsymbol{\tau}^M$ of the merged cut-cell K_i^M which span the local finite element space $\mathcal{S}(K_i^M)$ and $\mathcal{U}(K_i^M)$, respectively, are defined by

$$\upsilon^M = \upsilon \qquad \forall \mathbf{x} \in K_i^M \qquad \text{and} \qquad \upsilon^M = 0 \ \forall \mathbf{x} \notin K_i^M, \tag{3.137}$$

$$\boldsymbol{\tau}^M = \boldsymbol{\tau} \qquad \forall \mathbf{x} \in K_i^M \qquad \text{and} \qquad \boldsymbol{\tau}^M = \mathbf{0} \ \forall \mathbf{x} \notin K_i^M. \tag{3.138}$$

The shape functions υ and $\boldsymbol{\tau}$ are defined on the hexahedral element that contained the former neighbouring element K_i^N. Figure 3.5 shows an exemplary merged cut-cell K_i^M where the checkerboard pattern marks the hexahedral element on which the shape functions υ and $\boldsymbol{\tau}$ are defined. Figure 3.5 also shows that the support of the shape functions υ^M and $\boldsymbol{\tau}^M$ of K_i^M exceeds the hexahedral element on which they are defined. However, since the shape functions are polynomial functions, they are also well-defined outside of the red checkerboard pattern.

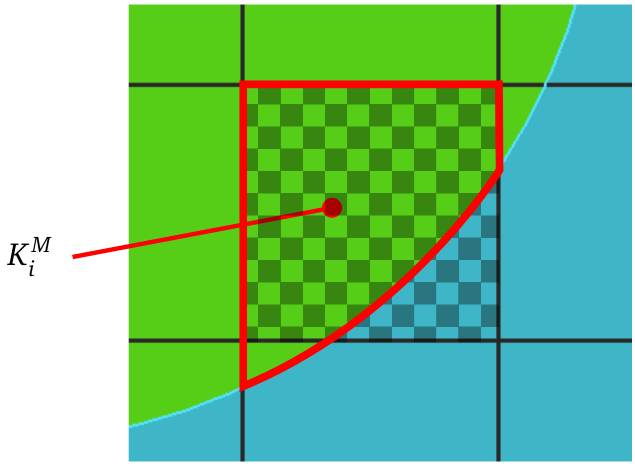

Figure 3.5.: Shape function support of the merged cut-cell K_i^M.

For the newly defined computational mesh Ω_h^{CM} and the finite element spaces V_h^{CM} and Σ_h^{CM} the bilinear form and the matrix formulation of the BCDG method with the additional *cell merging method* can be written as introduced in Section 3.3.2 and 3.2.4. Moreover, the system of equations is solved as shown in Section 3.2.4.

Figure 3.6 shows the numerical fluxes of an exemplary newly merged element K_i^M. The element K_i^M is connected by fluxes with all neighbouring elements that share a face with K_i^M. Note that some faces of the newly merged cell K_i^M are linked to two elements. For instance, face f of K_i^M is linked with K_j and K_l. Therefore, face f has two pairs of numerical fluxes, $\tilde{\phi}_{h|f_j}$, $\tilde{\mathbf{D}}_{h|f_j}$ and $\tilde{\phi}_{h|f_l}$, $\tilde{\mathbf{D}}_{h|f_l}$, since numerical flux terms are defined as integrals over the real faces f_j and f_l of the shared face $f = f_j \cup f_l$. The only flux that is dropped by the merging approach is the numerical flux term between K_i^C and K_i^N.

The geometry representation of the merged cells is obtained from OCCT and the numerical integration over these cells is performed using OCCT as shown in Section 3.3.3. Since we perform a face-wise matrix assembly, the evaluation of the integrals on merged cells can be easily conducted by changing the affiliation of the faces. More precisely, faces which are assigned to elements that are being merged are then assigned to the newly merged cell. This procedure saves computational costs. Numerical results for the *cell merging method* and the *adaptive approximation order method* are presented in Section 4.1.

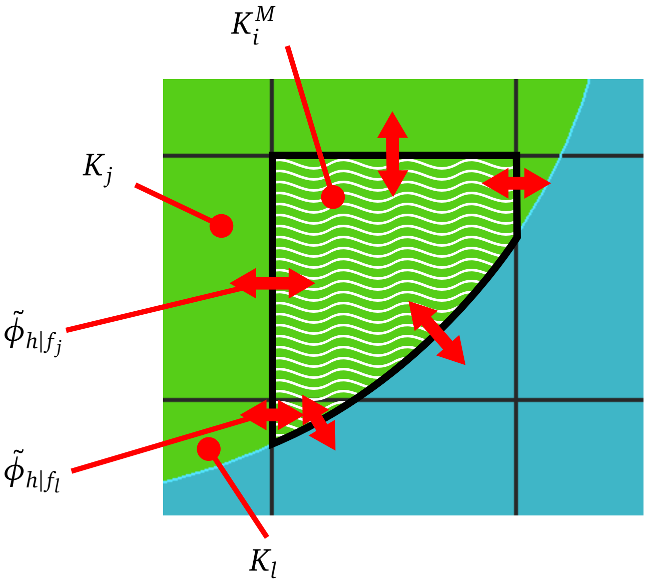

Figure 3.6.: Examplary numerical fluxes of the merged cut-cell K_i^M.

3.3.5 Discontinuous Galerkin (DG) Hybridization method

In this section we introduce the *DG hybridization method* for the *cut-cell discretization approach*. This hybridization approach combines the DG method with the FE method. As mentioned previously, one advantage of the DG method is its flexibility. For instance, it enables the natural embedding of the cut-cells near material interfaces or boundaries. However, compared to standard FE methods applied to meshes with curved elements, DG methods have a larger total number of DOF. The *DG hybridization method* is based on the observation that the potential and the electric field of the EQS field problems are continuously distributed in elements away form the interface or boundary. These elements are in general filled with a single material. Therefore, a continuous standard FE approximation would be more appropriate in these normal elements since similar results in terms of accuracy can be obtained combined with a substantially lower number of DOF.

One feature of the BCDG method makes the implementation of an hybridization method straightforward. The BCDG method uses shape functions which are chosen to be cell-wise identical with those of the FE method. This implies that the solution space of the FE method can be considered as a particular solution space of the DG method [33]. Thus, for a mesh with only normal hexahedral element it is possible to define two topological operators, **L** and **W**, which satisfy the following relations:

$$\boldsymbol{\phi}^{DG} = \mathbf{L}\boldsymbol{\phi}^{FE} \quad \text{and} \qquad \boldsymbol{\phi}^{FE} = \mathbf{W}\boldsymbol{\phi}^{DG}, \tag{3.139}$$

where $\boldsymbol{\phi}^{DG} \in \mathbb{R}^{N^{DG}}$ and $\boldsymbol{\phi}^{FE} \in \mathbb{R}^{N^{FE}}$ denote vectors which contain the coefficients $\phi_h^i(t)$ of the approximate DG and FE solution of the potential ϕ_h (see Section 3.2.2). The operator matrices $\mathbf{L} \in \mathbb{R}^{N^{DG} \times N^{FE}}$ and $\mathbf{W} \in \mathbb{R}^{N^{FE} \times N^{DG}}$ are purely topological, with **W** being the pseudo-inverse of **L** [33]. Note that the DG DOF are associated to each node, edge and face of each cell in the mesh whereas the FE DOF are associated to each node, edge and face of the mesh. Furthermore, the

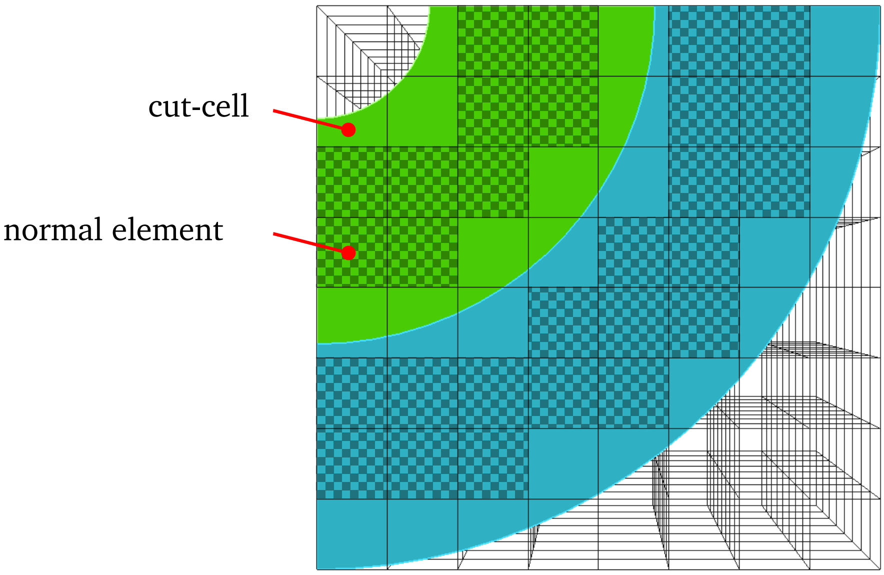

Figure 3.7.: Exemplary computational mesh Ω_h^C with normal elements and cut-cells.

numerical flux formulation in 3.2.3 allows that a particular potential solution of the DG method can be obtained by multiplying the FE DOF of each node, edge and face. Note that the FE DOF of each node, edge and face are multiplied by the number of elements the node, edge or face is associated to. Therefore, the operator matrix $\mathbf{L}$ contains components which are equal to 0 and 1 only. More precisely, for each FE node DOF the matrix components which refer to this node in the DG method framework are set to 1. The same can be done with all edge and face DOF. Since the operator matrix $\mathbf{W}$ is the pseudo-inverse of $\mathbf{L}$, the matrix components of a particular node DG DOF in the operator matrix $\mathbf{W}$ which refers to the node DOF of the FE method is set to $1/N_n$ where N_n denotes the number of element which are associated to this node. In a similar manner the matrix components of edge and face DOF are determined.

Next, relation (3.139) creating the link between FE and DG DOF in normal elements is used within the cut-cell discretization in order to reduce DOFs in normal elements. More precisely, relation (3.139) is applied only to the normal elements of our mesh Ω_h^C. The cut-cells $K_i^C \in \Omega_h^{CC}$ are excluded from this procedure. The new vector containing the coefficients of a hybrid potential solution $\boldsymbol{\phi}_h^{HY} \in \mathbb{R}^{N^{HY}}$ is defined by:

$$\boldsymbol{\phi}^{HY} = \begin{cases} (\phi_h^i(t))^{DG} & \text{in cut-cells,} \\ (\phi_h^i(t))^{FE} & \text{in normal elements,} \end{cases} \tag{3.140}$$

which has to satisfy

$$\boldsymbol{\phi}^{DG} = \mathbf{L}'\boldsymbol{\phi}^{HY} \quad \text{and} \quad \boldsymbol{\phi}^{HY} = \mathbf{W}'\boldsymbol{\phi}^{DG}, \tag{3.141}$$

where $\mathbf{L}' \in \mathbb{R}^{N^{DG} \times N^{HY}}$ and $\mathbf{W}' \in \mathbb{R}^{N^{HY} \times N^{DG}}$ are the cut-cell discretization operator matrices. The components of the operator matrices $\mathbf{L}'$ and $\mathbf{W}'$ corresponding to normal element DOFs are determined as presented above. However, the matrix components of $\mathbf{L}'$ and $\mathbf{W}'$ associated to the cut-cell DOFs are set to 1 on the diagonal and to 0 elsewhere. Therefore, relation (3.141) is always satisfied.

The matrix formulation of the hybrid scheme is obtained by entering (3.141) in (3.40) and multiplying with $\mathbf{W}'$. Thus, the matrix formulation of the hybrid scheme is given by

$$\mathbf{W}'\left(-\mathbf{G}^T\mathbf{I}_\epsilon\mathbf{M}^{-1}\mathbf{G}+\mathbf{S}\right)\mathbf{L}'\frac{d}{dt}\boldsymbol{\phi}^{HY}=\mathbf{W}'\left(\mathbf{G}^T\mathbf{I}_\kappa\mathbf{M}^{-1}\mathbf{G}-\mathbf{S}\mathbf{I}_{\kappa/\epsilon}\right)\mathbf{L}'\boldsymbol{\phi}^{HY}+\mathbf{W}'\mathbf{g}_B. \tag{3.142}$$

The hybrid matrix formulation is treated in the same way as the matrix formulation presented in (3.40).

The numerical results of the BCDG method combined with the *DG hybridization method* are presented in Section 4.1.6. It is shown that the number of DOF can be reduced substantially while the approximation errors are of the same order of magnitude as the approximation errors obtained without the additional *DG hybridization*.

Note that our method differs to the approach presented in [65] in the way the coupling of the FE and the DG method is performed. As already mentioned in Section 3.3.1, in [65] the domains where the LDG method and the conforming FE method is applied to have to be specified a priori. Then, the domains are discretized separately. In our case, the boundary conformal DG discretization is applied to the whole computational domain. Then, after the assembly routine, we conduct the DOF reduction using (3.140)-(3.141). Since we do not not know a priori how many cut-cells and how many normal elements a mesh contains, the coupled LDG-FE method introduced in [65] can not be used for the present implementation. Nevertheless, it may be possible to change the implementation in such a way that the number and location of the cut-cells is known a priori. If such a modification is performed, the coupled LDG-FE method in [65] could also be applied to the BCDG method. Finally, we would like to emphasize that the additional computational costs of the *DG hybridization* are caused by setting up the topological operator matrices and by multiplying the system of equation with the topological operator matrices. Since these operations can be performed efficiently, the additional computational costs of the DG hybridizations are limited and are clearly outweighed by the benefit of having less DOF.

3.3.6 Numerical properties of the BCDG method

In Section 3.2.6, we have shown that the DG method satisfies the four most important properties for numerical methods, namely stability, consistency, conservativeness and convergence. In this section, we briefly state that the BCDG method also satisfies all four properties for the following reasons: First, the cut-cells are considered as independent elements in the BCDG method which are connected by the numerical LDG fluxes just like normal elements in the mesh. Second, the local finite element spaces $\mathcal{S}(K_i^C)$ and $\mathcal{U}(K_i^C)$ of cut-cells are also polynomial spaces. Therefore, these spaces satisfy condition (3.68) which is given by

$$\nabla\mathcal{S}(K_i^C)\subset\mathcal{U}(K_i^C).$$

This implies that the BCDG method meets all required conditions for the property proofs presented in Section 3.2.6. Since the properties can be derived in exactly the same way as presented in Section 3.2.6, we refer to Section 3.2.6 for the derivation of the properties.

3.4 Discretization of time

The spatial and the time discretization of the electro-quasistatic Maxwell's equations is performed separately in this work. As mentioned previously, this kind of separation is referred to as method of lines [86]. In the previous sections, we have introduced the DG method and the BCDG method which are applied to the spatial derivatives of the EQS Maxwell's equations. Both the DG and the BCDG method transform the EQS Maxwell's equations to a semi-discrete weak DG formulation which still contains the time derivative. In this section, the discretization of the time derivative in the semi-discrete weak DG formulation is described. First, the semi-discrete weak DG formulation is introduced in a form typically used to present numerical methods for systems of ODEs. Second, the Θ-method used to discretize the time derivative is presented. Finally, the properties of the Θ-method are discussed.

Since the semi-discrete weak DG formulation of the EQS Maxwell's equations in (3.33)-(3.39) contains a time derivative, it belongs to the class of first order ordinary differential equations (ODEs). Note that systems of first order ODEs are usually presented in the explicit form. To derive the explicit form of the system of ODEs we use the matrix formulation[4] of the semi-discrete weak DG formulation. This matrix formulation can be written as follows:

$$\mathbf{M}'\frac{d}{dt}\boldsymbol{\phi}(t) = \mathbf{K}'\boldsymbol{\phi}(t) + \mathbf{g}_B, \tag{3.143}$$

with

$$\begin{aligned} \mathbf{M}' &= \left(-\mathbf{G}^T\mathbf{I}_\epsilon\mathbf{M}^{-1}\mathbf{G} + \mathbf{S}\right), & (3.144)\\ \mathbf{K}' &= \left(\mathbf{G}^T\mathbf{I}_\kappa\mathbf{M}^{-1}\mathbf{G} - \mathbf{S}\mathbf{I}_{\kappa/\epsilon}\right). & (3.145)\end{aligned}$$

where the vector $\boldsymbol{\phi}(t)$ contains the coefficients $\phi_h^i(t)$ of the potential solution $\phi_h(x,t)$ in (3.9). The matrices $\mathbf{M}'$ and $\mathbf{K}'$ are square matrices with $\mathbf{M}' \in \mathbb{R}^{N\times N}$ and $\mathbf{K}' \in \mathbb{R}^{N\times N}$. Note that the matrix $\mathbf{M}'$ is regular since we always use permittivity values $\epsilon \neq 0$ in our EQS approximation.
Moreover, the system of equations in (3.144) is a system of linear ordinary differential equations [11]. Since the matrix M is invertible, the system of first order ODEs can be written in the explicit form [78]:

$$\frac{d}{dt}\boldsymbol{\phi}(t) = \mathbf{f}(\boldsymbol{\phi}(t), t), \quad \text{with} \quad \boldsymbol{\phi}(t_0) = \boldsymbol{\phi}_0, \tag{3.146}$$

where $\mathbf{f} = \mathbf{M}'^{-1}\mathbf{K}'\boldsymbol{\phi}(t) + \mathbf{M}'^{-1}\mathbf{g}_B$ and $\boldsymbol{\phi}(t_0)$ is the initial condition of the ordinary differential equation. Note that the solution of (3.146) is only uniquely defined if we are given well-defined initial condition values $\boldsymbol{\phi}_0$. Hence, the system of ordinary differential equations in (3.146) is also called initial value problem. Furthermore, it was shown that for the system of equations in (3.146) a uniquely defined solution exists if the vector function $\mathbf{f}$ is Lipschitz-continuous (see, e.g. [78]). Therefore, we assume that the Lipschitz-continuity conditions are always satisfied.

[4] Note that the bilinear form and the matrix formulation are equivalent. However, applying the standard numerical methods for ODEs to the bilinear form of the weak DG formulation is much more complicated. Therefore, we prefer to present the matrix formulation of the semi-discrete weak DG formulation.

The classical numerical methods for ODEs operate on a particular time grid. Therefore, the time domain $[t_0, t_N]$ is split into n equidistant time intervals $\Delta t = (t_N - t_0)/n$ [77]. The solution of (3.146) is evaluated at the following points in time:

$$t^n = t_0 + n\Delta t,\ n = 0, 1, \ldots, N. \tag{3.147}$$

These points are also called grid points and the time interval Δt is often denoted as grid size [77].

In most applications either *one-step methods* or *multistep methods* are used to solve initial value problems as presented in (3.146). *One-step methods* calculate the solution $\boldsymbol{\phi}^{n+1}$ at t^{n+1} based on the solution $\boldsymbol{\phi}^n = \boldsymbol{\phi}(t^n)$ calculated at time t^n, whereas *multistep methods* use several previously computed solutions [78].

In this work, we use the Θ-method which belongs to the class of *one-step methods*. In general, *one-step* methods are written as follows:

$$\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^n + \Delta t \Phi(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n, h), \tag{3.148}$$

where $\Phi(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n, h)$ is the increment function [78].

Note that *one-step methods* can be further divided into two classes:

- *Explicit methods*: The calculation of the new solution $\boldsymbol{\phi}^{n+1}$ includes only solution information available at time t^n. In other words, the increment function is given as $\Phi(\boldsymbol{\phi}^n, h)$.
- *Implicit methods*: The right side of (3.148) contains also the (unknown) new solution $\boldsymbol{\phi}^{n+1}$. Therefore, the increment functions depends also on $\boldsymbol{\phi}^{n+1}$, i.e. $\Phi(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n, h)$.

3.4.1 The Θ-method

In our implementation, we use the simplest type of *one-step* methods, the so-called Θ-method which is given by the following equation (see, e.g. [69]):

$$\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^n + (1 - \Theta)\Delta t\ \mathbf{f}(\boldsymbol{\phi}^n, t^n) + \Theta\Delta t\ \mathbf{f}(\boldsymbol{\phi}^{n+1}, t^{n+1}), \tag{3.149}$$

with the parameter $\Theta \in [0, 1]$. The increment function of the Θ-method reads

$$\Phi(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n, h) = (1 - \Theta)\ \mathbf{f}(\boldsymbol{\phi}^n, t^n) + \Theta\ \mathbf{f}(\boldsymbol{\phi}^{n+1}, t^{n+1}). \tag{3.150}$$

Note that the parameter Θ governs whether the Θ-method is considered as an explicit or implicit method. For $\Theta = 0$ the method belongs to the class of explicit methods, whereas for $\Theta > 0$ it belongs to the class of implicit methods. Furthermore, for particular Θ parameter values, the method is equivalent to the following well-known methods:

- *Backward Euler* or *implicit Euler* method for $\Theta = 1$
- *Crank-Nicolson* method or *trapezoidal rule* for $\Theta = 1/2$
- *Forward Euler* or *explicit Euler* method for $\Theta = 0$

3.4.1.1 Properties of the Θ-method for the parameter values $\Theta = \{0, 1/2, 1\}$

In the following we show the consistency, stability and convergence of the Θ-method.
The consistency of the Θ-method can be shown using the *local discretization error* $\tau(\Delta t)$ which is defined by

$$\tau(\Delta t) = \frac{\mathbf{z}(t_0 + \Delta t) - \mathbf{z}_0}{\Delta t} - \Phi(\boldsymbol{\phi}^1, \mathbf{z}_0, h), \tag{3.151}$$

where Φ is the *increment function* specified in (3.148) and $\boldsymbol{\phi}^1$ the approximate solution of the Θ-method at t^1 [73]. Furthermore, $\mathbf{z}$ denotes the exact solution of the system of ODEs in (3.146) for the given initial conditions $\mathbf{z}_0 = \mathbf{z}(t_0)$. In [73] it is shown that the Θ-method is consistent since it satisfies

$$||\tau(\Delta t)|| \leq \gamma(\Delta t) \text{ with } \lim_{\Delta t \to 0} \gamma(\Delta t) = 0. \tag{3.152}$$

This holds true for all parameter values $\Theta \in [0, 1]$.
Furthermore, the local discretization error defines the consistency order of the system which, in turn, specifies the approximation quality of each approximation step. The consistency order p is given by

$$||\tau(\Delta t)|| = \mathcal{O}((\Delta t)^p), \tag{3.153}$$

where $\mathcal{O}(\cdot)$ is the Landau symbol [73].
Next, we introduce the global discretization error which is needed to show the convergence of the Θ-method. The *global discretization error* $e_n(T)$ of the method is defined by

$$e_n(T) = \mathbf{z}(T) - \boldsymbol{\phi}^n \quad \text{with} \quad T = t^n, n = 1, ..., N. \tag{3.154}$$

In [73] it is shown that the Θ-method converges if $e_n(T)$ satisfies

$$\lim_{n \to \infty} ||e_n(T)|| = 0. \tag{3.155}$$

This condition is fulfilled for all parameter values $\Theta \in [0, 1]$ if the grid size is chosen sufficiently small [73]. Furthermore, the convergence definition shows that the consistency of the Θ-method implies the convergence of the Θ-method.
The global order of the Θ-method is defined by

$$||e_n(T)|| = \mathcal{O}((\Delta t)^p), \tag{3.156}$$

when the function $\mathbf{f}$ is Lipschitz-continuous [73]. The convergence order p depends on the Θ value: For $\Theta \in [0, 1] \setminus \{1/2\}$ the Θ-method is of global order $p = 1$. For $\Theta = 1/2$ the method is of order $p = 2$ [78]. Furthermore, it can be shown that the consistency order and the global order of the method are identical for all $\Theta \in [0, 1]$ [78].
Next, we would like to discuss the stability property of the Θ-method. The stability of the Θ-method is closely linked with the stiffness phenomena of systems of ODEs. A system of ODEs is called stiff if the following two conditions hold: First, explicit numerical methods converge

only for very small grid sizes Δt. Second, implicit methods converge although much higher grid sizes are used. In other words, stiff ODEs are ODEs where implicit methods perform much better than explicit methods [73]. For linear systems of ODEs the stiffness behaviour may be detected by evaluating the eigenvalues: A system of ODEs is usually considered as stiff if the system contain some very high negative as well as some very low negative eigenvalues, i.e. the ratio between the highest and the lowest negative eigenvalue is very high.

Since the semi-discrete weak DG formulation of one of our numerical examples can be considered as a stiff system of ODEs, we introduce in the following a term which captures the stability of numerical schemes on stiff ODEs. This stability term is called *A-stability* (*absolute stability*) and is based on the following *test equation*[5] [78]:

$$\frac{d}{dt}y = \lambda y \quad \text{with} \quad \lambda \in \mathbb{C},\, Re\lambda \leq 0. \tag{3.157}$$

The exact solution of the *test equation* converges to 0 for $t \to \infty$. In addition, the detailed definition of the A-stability requires that the Θ-method is written in the following form [73]:

$$y^{i+1} = y^i + \Delta t\ \Phi(t) = R(z)y^i \quad \text{with} \quad R(z) = \frac{1+(1-\Theta)z}{1-\Theta z},\, z = \lambda\Delta t. \tag{3.158}$$

The function R determines the stability behaviour of the Θ-method. Considering (3.157) and (3.158), a numerical method is denoted as *A-stable* if

$$|R(z)| \leq 1 \forall z \in \mathbb{C}^- := \{z \in \mathbb{C} : Re\ z \leq 0\}. \tag{3.159}$$

Using definition (3.159), it can be shown that the Θ-method is *A-stable* for $\Theta = 1/2$ (Crank-Nicolson method) and for $\Theta = 1$. However, the Θ-method for $\Theta = 0$ (explicit Euler method) is not *A-stable* [78].

Furthermore, definition (3.159) and (3.159) indicate which time step sizes Δt result in stable results. If the Θ-method is applied to our system of linear equations $d/dt\mathbf{y} = \mathbf{Ay}$ as given in (3.146), we have to ensure that we choose a time step Δt for which the highest absolute eigenvalue $\lambda_m = \max_i \lambda_i$ of $\mathbf{A}$ satisfies the following relation:

$$|R(\Delta t \lambda_m)| \leq 1. \tag{3.160}$$

If this condition is fulfilled for the chosen Δt value, the Θ-method yields stable results.

In addition to *A-stability*, a further stability condition, the *L-Stability condition*, provides further information about the behaviour of the numerical method on stiff differential equations. A method is denoted as L-stable, if the method is A-stable and if $R(z)$ satisfies

$$\lim_{z\to\infty} R(z) = 0. \tag{3.161}$$

In [73] it is shown that the implicit Euler method is *L-stable*, whereas the Crank-Nicolson is not *L-stable*. In general, *L-stable* methods have some damping effects on the oscillations for higher Δt values. This effect is called numerical damping [73]. In some cases this is desirable. For instance, if a *L-stable* method is applied to the test equation in (3.157) having the eigenvalues

[5] The test equation goes back to Dahlquist (1963).

$\lambda_i < 0$ for all $i = 1, ..., n$. In this case, the damping effect of the *L-stable* methods correspond to the asymptotic behaviour of the solution of the test equation. By contrast, if a *L-stable* method is applied to a system of linear ODEs having some eigenvalues λ_i with $Re\lambda_i = 0$, the *L-stable* method might have damping effects on the resulting free oscillations of the system. In this case, numerical methods that are not *L-stable* are preferred, since they do not show damping effects. Finally, we would like to point out that implicit methods involve higher computational costs since a system of equation has to be solved at each point in time. However, these additional costs are justified if the system of equations shows certain stiffness behaviour. In this case the time step Δt can be set to a significantly higher value for implicit methods which compensates the additional costs [78].

The numerical results of the BCDG method for EQS field problems on the application example with non-moving material distributions are obtained using the Crank-Nicolson method ($\Theta = 1/2$). We used the Crank-Nicolson method since it is a second-order method. However, the EBCDG method is applied using always the implicit Euler method ($\Theta = 1$). As we will explain in Section 3.6, the EBCDG method employs a projection scheme which requires the use of the implicit Euler method. Moreover, as we will show in Section 4, the EBCDG method is applied to a application example which shows a stiffness behaviour, further justifying the use of the implicit Euler method.

3.4.2 Application of the Θ-method to the semi-discrete weak DG formulation

In this section, we briefly introduce the system of linear equations we obtain from applying the Θ-method to the semi-discrete weak DG formulation of the DG and the BCDG method. We show the resulting system of linear equations for both the matrix formulation and the bilinear form of the semi-discrete weak DG formulation, since both formulations are used in the following.

Applying the Θ-method to the matrix formulation of the semi-discrete weak DG formulation in (3.109) or in (3.40) yields the following system of linear equations:

$$\begin{aligned}\left(-\mathbf{G}^T\mathbf{I}_{(\epsilon+\Theta\Delta t\,\kappa)}\mathbf{M}^{-1}\mathbf{G}+\mathbf{S}\mathbf{I}_{(1+\Theta\Delta t\,\kappa/\epsilon)}\right)\boldsymbol{\phi}^{n+1} &= \\ &\left(-\mathbf{G}^T\mathbf{I}_{(\epsilon-(1-\Theta)\Delta t\,\kappa)}\mathbf{M}^{-1}\mathbf{G} \;+\; \mathbf{S}\mathbf{I}_{(1-(1-\Theta)\Delta t\,\kappa/\epsilon)}\right)\boldsymbol{\phi}^{n}+\mathbf{g}_B,\end{aligned} \tag{3.162}$$

where the matrices $\mathbf{M}, \mathbf{G}, \mathbf{S}$ and the vector $\mathbf{g}_B$ are defined as in (3.35)-(3.39). At each point in time t^{n+1}, equation (3.162) is solved for the potential solution $\boldsymbol{\phi}^{n+1}$, using either a direct or a iterative solver for systems of linear equations. The reader is referred to Section 3.5 for more details on the solvers used.

The flux density solution vector $\mathbf{d}^{n+1}$ containing the flux density coefficients $D_h^i(t^{n+1})$ is obtained by entering $\boldsymbol{\phi}^{n+1}$ in (3.33) for each point in time:

$$\mathbf{d}^{n+1} = \mathbf{M}^{-1}\mathbf{I}_\epsilon\mathbf{G}\boldsymbol{\phi}^{n+1} + \mathbf{M}^{-1}\mathbf{I}_\epsilon\mathbf{f}. \tag{3.163}$$

Next, we introduce the formulation of the system of linear equations which is obtained by applying the Θ-method to the bilinear form of the semi-discrete weak DG formulation. As mentioned previously, the bilinear form and the matrix formulation of the weak DG formulation are equivalent. However, we also present the bilinear formulation, as it is used to introduce the EBCDG method in Section 3.6. The Θ-method applied to the bilinear form presented in (3.105)-(3.106) for the BCDG method and in (3.25)-(3.31) for the DG method yields the following

discrete weak DG formulation:
Find ϕ_h and $\mathbf{D}_h$ such that the following equations are satisfied at time t^{n+1} with $n = 1, ..., N$:

$$\begin{aligned} a\left(\frac{1}{\epsilon}\mathbf{D}_h^{n+1}, \boldsymbol{\tau}\right) &= b\left(\phi_h^{n+1}, \boldsymbol{\tau}\right) + f\left(\boldsymbol{\tau}\right) && (3.164) \\ -b\left(\upsilon, (1+\Theta\Delta t\, \kappa/\epsilon)\mathbf{D}_h^{n+1}\right) &+ s\left(\upsilon^{n+1}, (1+\Theta\Delta t\, \kappa/\epsilon)\phi_h^{n+1}\right) = && (3.165) \\ -b\left(\upsilon, (1-(1-\Theta)\Delta t\, \kappa/\epsilon)\mathbf{D}_h^{n}\right) &+ s\left(\upsilon, (1-(1-\Theta)\Delta t\, \kappa/\epsilon)\phi_h\right) + g\left(\upsilon\right), \end{aligned}$$

$\forall \upsilon \in V_h, \forall \boldsymbol{\tau} \in \Sigma_h$, where a, b, s, f, g are defined as shown in (3.27)-(3.31). The resulting system of linear equations is solved either with the direct or with the iterative solver presented in the next section.

3.5 Numerical treatment of the system of linear equations

The numerical solutions of the DG and the BCDG method are obtained by solving the discrete weak DG formulations for the solution coefficients. In this section we briefly present both the direct and the iterative solver used to solve the discrete weak DG formulations.

As shown in Section 3.4.2, applying the Θ-method to the semi-discrete weak DG formulation results in systems of linear equations which have to be solved at each point in time. Furthermore, as we have shown in Section 3.3.2 and 3.2.5, applying the DG or the BCDG method to the ES approximation of the Maxwell's equations yields a system of linear equations which also has to be solved for the solution coefficients.
Obviously, such a system of linear equations can be written in the form

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{b}, \tag{3.166}$$

where A denotes the system matrix and b the right-hand side vector. The vector $\boldsymbol{\phi}$ contains the coefficients of the potential solution ϕ_h. We refer to Section 3.4.2, Section 3.3.2 and Section 3.2.5 for detailed information on the system of linear equations of the DG and the BCDG method for EQS and ES field problems.
In this work, the systems of linear equations are solved using either the direct solver *SuperLU_DIST* or the iterative *conjugate gradient (CG)* method, both provided by PETSc [3].
The *SuperLU_DIST* solver is a scalable distributed-memory sparse solver for large systems of linear equations. It is based on a Gaussian elimination. However, it uses a innovative static pivoting strategy which enables a fast parallelization of the solver [54].
The *CG method* belongs to the class of Krylov Subspace methods and iteratively solves the system of linear equations until a approximate solution is found which satisfies a pre-specified residuum accuracy. We use the *CG method* in combination with the *incomplete LU (ILU) preconditioner* [3].
The *CG method* has the advantage that it can be applied to very large sparse systems of linear equations that cannot be handled by direct solvers. The reader is referred to the literature for further detail on both solvers (see, e.g. [3], [85], [74]).

3.6 Boundary conformal Discontinuous Galerkin method on moving material distributions

In Section 3.3, we introduced the DG method with the additional boundary conformal approach to solve electro-quasistatic applications with non-moving material distributions. However, in certain real world applications materials move over time. A well-known example is water droplets oscillating on the insulation layer of high voltage insulators. To apply the BCDG method to field problems consisting of materials that move over time, we need to take into account the time variable within the cut-cell discretization approach. In this section, we therefore present an extension of the BCDG method which can be applied to field problems with moving materials. This extension of the BCDG method is referred to as EBCDG method in the following.

Most approaches developed for field problems with moving material distributions are based on meshes that are fitted to the boundary and, furthermore, use a remeshing or mesh adaption technique to fit the mesh to the material boundary at any point in time. For instance, in [75] a method based on a finite element scheme on curved conforming meshes is presented for the water droplet insulator application example (see, e.g. Chapter 1). In contrast to methods using a remeshing or mesh adaption strategies, the EBCDG method is applied to a single, time constant hexahedral mesh. Instead of adapting the mesh to the moving material boundaries, the EBCDG method adapts the *cut-cell discretization approach* to the moving material boundaries to obtain sufficiently accurate results. The *cut-cell discretization approach* of the BCDG method is designed to accurately treat arbitrarily shaped material interfaces on hexahedral meshes. Therefore, the adaption of the *cut-cell discretization approach* to moving material interfaces can be easily performed by applying the cut-cell discretization to the material interfaces at each point in time. This implies that only the cut-cells have to be re-evaluated at each point in time. As the number of cut-cells in a mesh is generally much smaller than the number of normal elements, the application of the cut-cell discretization has the potential to be more efficient than a remeshing or mesh adaption strategy.

Applying the cut-cell discretization to materials that move over time on a fixed mesh results in cut-cell domains which change their volume from one point in time to the next point in time. Furthermore, the cut-cells are treated at each point in time as independent elements within the framework as presented in Section 3.3. This implies that each cut-cell has its own set of approximation functions and DOF. However, in contrast to the BCDG method in Section 3.3, the approximation and test functions of the EBCDG method become time dependent. In the following, we use the notation n and $n+1$ as introduced in Section 3.4 to indicate the variables at time t^n and t^{n+1}. For instance, the material domain of an exemplary cut-cell is denoted as $K_i^{C,n}$ at time t^n and as $K_i^{C,n+1}$ at time t^{n+1}. Furthermore, the computational mesh including normal elements and cut-cells at time t^n is defined by $\Omega_h^{C,n} = \Omega_h^{CC,n} \cup \Omega_h^{N,n}$ where $\Omega_h^{CC,n}$ and $\Omega_h^{N,n}$ represent the set of cut-cells and the set of normal elements at time t^n, respectively. After the cut-cell discretization is again applied to the material domain at the next point in time, t^{n+1}, the computational mesh is given by $\Omega_h^{C,n+1} = \Omega_h^{CC,n+1} \cup \Omega_h^{N,n+1}$, where $\Omega_h^{CC,n+1}$ and $\Omega_h^{N,n+1}$ are the set of cut-cells and normal elements at time t^{n+1}, respectively. As the material interface or boundary moves over time, the number of cut-cells or normal elements in the computational mesh might change during the simulation. For example, from time t^n to time t^{n+1} the material distribution may change in such a way that material interfaces or boundaries enter elements which have been considered as normal elements at time t^n. Such an element is then split into two independent cut-cells. Obviously, the opposite may happen as well. Elements considered as cut-cells at time t^n may become normal elements at time t^{n+1}.

Next, we introduce the finite element spaces of the approximate solutions of the EBCDG method for a given point in time. The approximate solutions ϕ_h and $\mathbf{D}_h$ at time t^{n+1} are denoted as $\phi_h^{n+1} = \phi_h(t^{n+1}, \mathbf{x})$ and $\mathbf{D}_h^{n+1} = \mathbf{D}(t^{n+1}, \mathbf{x})$. Since the approximation and test functions change with time, the global finite element spaces V_h and Σ_h become time dependent as well. Therefore, the approximate solutions ϕ_h^{n+1} and $\mathbf{D}_h^{n+1}$ which we need to determine at time t^{n+1} belong to the finite element spaces V_h^{n+1} and Σ_h^{n+1} given by

$$\begin{aligned} V_h^{n+1} &= \{v^{n+1} \in L^2(\Omega) : v^{n+1}|_{K_i^{n+1}} \in \mathcal{S}^{n+1}(K_i^{n+1}) \forall K_i^{n+1} \in \Omega_h^{C,n+1}\}, && (3.167) \\ \Sigma_h^{n+1} &= \{\boldsymbol{\tau}^{n+1} \in [L^2(\Omega)]^3 : \boldsymbol{\tau}^{n+1}|_{K_i^{n+1}} \in \mathcal{U}^{n+1}(K_i^{n+1}) \forall K_i^{n+1} \in \Omega_h^{C,n+1}\}, && (3.168) \end{aligned}$$

where $\mathcal{S}^{n+1}(K_i^{n+1})$ and $\mathcal{U}^{n+1}(K_i^{n+1})$ are the local finite element spaces of element K_i^{n+1} at time t^{n+1}. These local finite element spaces are spanned by the shape functions $v^{n+1} \in \mathcal{S}^{n+1}(K_i^{n+1})$ and $\boldsymbol{\tau}^{n+1} \in \mathcal{U}^{n+1}(K_i^{n+1})$.

As shown in Section 3.3, the local finite element spaces of cut-cells $K_i^{C,n+1} \in \Omega_h^{CC,n+1}$ differ from the local finite element spaces of normal hexahedral elements $K_i^{n+1} \in \Omega_h^{N,n+1}$. In cut-cells $K_i^{C,n+1} \in \Omega_h^{CC,n+1}$, the shape functions $v^{n+1} \in \mathcal{S}^{n+1}(K_i^{C,n+1})$ and $\boldsymbol{\tau}^{n+1} \in \mathcal{U}^{n+1}(K_i^{C,n+1})$ are defined as

$$\begin{aligned} v^{n+1} &= v \qquad \forall \mathbf{x} \in K_i^{C,n+1}, && (3.169) \\ v^{n+1} &= 0 \qquad \forall \mathbf{x} \in K \setminus K_i^{C,n+1}, && (3.170) \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{\tau}^{n+1} &= \boldsymbol{\tau} \qquad \forall \mathbf{x} \in K_i^{C,n+1}, && (3.171) \\ \boldsymbol{\tau}^{n+1} &= \mathbf{0} \qquad \forall \mathbf{x} \in K \setminus K_i^{C,n+1}, && (3.172) \end{aligned}$$

respectively. The functions v and $\boldsymbol{\tau}$ represent shape functions which are defined on the hexahedral element K containing the cut-cell domain $K_i^{C,n+1}$. This implies that the shape functions v^{n+1} and $\boldsymbol{\tau}^{n+1}$ change from time t^n to t^{n+1} because the cut-cell domain changes. However, the shape functions v and $\boldsymbol{\tau}$ defined on the hexahedral element remain unchanged.

Since normal elements are not intersected by a moving material interfaces or boundaries, the shape functions $v^{n+1} \in \mathcal{S}^{n+1}(K_i^{n+1})$ and $\boldsymbol{\tau}^{n+1} \in \mathcal{U}^{n+1}(K_i^{n+1})$ of normal elements $K_i^{n+1} \in \Omega_h^{N,n+1}$ are defined as shown in Section 3.2.7.

As discussed in Section 3.3, the approximate solutions $\phi_h^{n+1} \in V_h^{n+1}$ and $\mathbf{D}_h^{n+1} \in \Sigma_h^{n+1}$ for an exemplary element $K_i^{n+1} \in \Omega_h^{C,n+1}$ can be expressed in terms of shape functions:

$$\begin{aligned} \phi_{h,K_i^{n+1}}^{n+1} &= \sum_{i=j}^{N} \phi_{h,j}^{n+1} v_j^{n+1}(\mathbf{x}), && (3.173) \\ \mathbf{D}_{h,K_i^{n+1}}^{n+1} &= \sum_{i=j}^{N} \mathbf{D}_{h,j}^{n+1} \boldsymbol{\tau}_j^{n+1}(\mathbf{x}). && (3.174) \end{aligned}$$

The direct sum over all cells yields

$$\phi_h^{n+1}(\mathbf{x}) = \bigoplus_{i=1}^{N^{C,n+1}} \phi_{h,K_i^{n+1}}^{n+1}(\mathbf{x}), \tag{3.175}$$

$$\mathbf{D}_h^{n+1}(\mathbf{x}) = \bigoplus_{i=1}^{N^{C,n+1}} \mathbf{D}_{h,K_i^{n+1}}^{n+1}(\mathbf{x}). \tag{3.176}$$

We have just introduced the approximate solution ϕ_h^{n+1} and $\mathbf{D}_h^{n+1}$. Next, we introduce the semi-discrete weak DG formulation for EQS field problems with moving material distributions.

3.6.1 Bilinear form of the EBCDG method

When deriving the bilinear form of the EBCDG method, we follow closely the derivation of the weak DG formulation shown in Section 3.2 and 3.3. Note that with moving material distributions the test functions, $\upsilon^{n+1} \in V_h^{n+1}$ and $\boldsymbol{\tau}^{n+1} \in \Sigma_h^{n+1}$, and the approximate solutions, $\phi_h^{n+1} \in V_h^{n+1}$ and $\mathbf{D}_h^{n+1} \in \Sigma_h^{n+1}$ are time dependent. As in the case of the BCDG method, the bilinear form of the EBCDG method can be derived by multiplying the mixed formulation of the EQS Maxwell's equations by the test functions $\upsilon^{n+1} \in V_h^{n+1}$ and $\boldsymbol{\tau}^{n+1} \in \Sigma_h^{n+1}$, integrating over the computational domain Ω_h, specifying the numerical fluxes and summing over all elements. Thus, the bilinear form of the EBCDG method reads:
Find $\mathbf{D}_h(\mathbf{x}, t)$, $\phi_h(\mathbf{x}, t)$ such that

$$a\left(\frac{1}{\epsilon}\mathbf{D}_h, \boldsymbol{\tau}^{n+1}\right) = b\left(\phi_h, \boldsymbol{\tau}^{n+1}\right) + f\left(\boldsymbol{\tau}^{n+1}\right), \tag{3.177}$$

$$-b\left(\upsilon^{n+1}, (\frac{\partial}{\partial t}\mathbf{D}_h + \frac{\kappa}{\epsilon}\mathbf{D}_h)\right) + s\left(\upsilon^{n+1}, (\frac{\partial}{\partial t}\phi_h + \frac{\kappa}{\epsilon}\phi_h)\right) = g(\upsilon^{n+1}), \tag{3.178}$$

$\forall \upsilon^{n+1} \in V_h^{n+1}, \forall \boldsymbol{\tau}^{n+1} \in \Sigma_h^{n+1}$, where a, b, s, f, g are defined as shown in (3.27)-(3.31).
Next, the Θ-method with $\Theta = 1$ is applied to (3.177)-(3.178). The resulting discrete weak DG formulation reads:
Find $\phi_h^{n+1} \in V_h^{n+1}$ and $\mathbf{D}_h^{n+1} \in \Sigma_h^{n+1}$ such that the following equations are satisfied at each point in time t^{n+1}:

$$a\left(\frac{1}{\epsilon}\mathbf{D}_h^{n+1}, \boldsymbol{\tau}^{n+1}\right) = b\left(\phi_h^{n+1}, \boldsymbol{\tau}^{n+1}\right) + f\left(\boldsymbol{\tau}^{n+1}\right), \tag{3.179}$$

$$\begin{aligned} -b\left(\upsilon^{n+1}, (1 + \Delta t\,\kappa/\epsilon)\mathbf{D}_h^{n+1}\right) &+ s\left(\upsilon^{n+1}, (1 + \Delta t\,\kappa/\epsilon)\phi_h^{n+1}\right) = \\ &- b\left(\upsilon^{n+1}, \mathbf{D}_h^n\right) + s\left(\upsilon^{n+1}, \phi_h^n\right) + g\left(\upsilon^{n+1}\right), \end{aligned} \tag{3.180}$$

$\forall \upsilon^{n+1} \in V_h^{n+1}$ and $\forall \boldsymbol{\tau}^{n+1} \in \Sigma_h^{n+1}$. The approximate solutions $\mathbf{D}_h^{n+1}$ and ϕ_h^{n+1} for each point in time t^{n+1} are obtained by solving (3.179)-(3.180) for $\mathbf{D}_h^{n+1}$ and ϕ_h^{n+1}. In this case, we use the direct solver introduced in Section 3.3.4.1 to solve the system of linear equations in (3.179)-(3.180). Furthermore, the *adaptive approximation order method* is used with the EBCDG method to avoid ill-conditioning. Since the *adaptive approximation order method* only lowers the approximation order in geometrically small cut-cells, it is easy to implement it in the EBCDG method and doesn't change the framework of the EBCDG method.

However, solving (3.179)-(3.180) is not straightforward, as the right-hand side term of (3.180) contains approximation and test functions which are defined on different computational meshes. The right-hand side term[6] of (3.180) is given by

$$-b\left(v^{n+1},\mathbf{D}_h^n\right)+s\left(v^{n+1},\phi_h^n\right)+g\left(v^{n+1}\right), \tag{3.181}$$

where $b(v^{n+1},\mathbf{D}_h^n)$ and $s(v^{n+1},\phi_h^n)$ are defined by

$$b(v^{n+1},\mathbf{D}_h^n) = \sum_{K\in\Omega_h}\int_K \nabla\cdot\mathbf{D}_h^n v^{n+1}dV - \int_{\mathcal{F}_0}\{v^{n+1}\}[\![\mathbf{D}_h^n]\!]dA, \tag{3.182}$$

$$s(v^{n+1},\phi_h^n) = \int_{\mathcal{F}_0} C_{11}[\![\phi_h^n]\!]\cdot[\![v^{n+1}]\!]dA+\int_{\mathcal{F}_D} C_{11}v^{n+1}\phi_h^n dA, \tag{3.183}$$

respectively (see Section 3.3.2). Note that the test functions v^{n+1} in (3.181) belong to the finite element space V_h^{n+1} and the approximate solutions ϕ_h^n and $\mathbf{D}_h^n$ belong to V_h^n and Σ_h^n. Since the cut-cell domains change from t^n to t^{n+1}, the finite element spaces , V_h^n, Σ_h^n and V_h^{n+1}, Σ_h^{n+1}, differ with respect to the support of the approximation functions in the cut-cells. Moreover, the number of mesh elements might also change from t^n to t^{n+1}. This implies that the direct evaluation of (3.181) is not feasible. In the next section we present in detail how an approximation of (3.181) can be derived.

3.6.2 Evaluation of bilinear forms containing test and approximation functions which belong to different finite element spaces

In this section, we discuss the evaluation of the integral terms of $b(v^{n+1},\mathbf{D}_h^n)$ and $s(v^{n+1},\phi_h^n)$ in (3.181). As mentioned previously, the shape functions of normal elements are defined as shown in 3.3 and do not change from t^n to t^{n+1}, provided that no material interface enters the normal element. Therefore, the integral terms of $b(v^{n+1},\mathbf{D}_h^n)$ and $s(v^{n+1},\phi_h^n)$ in (3.181) on normal elements can be evaluated as shown in Section 3.3. Note that we assume computational domains with moving material interfaces such as oscillating water droplets surrounded by air. This implies that the boundary faces are always faces of normal hexahedral elements. Since the shape functions of normal elements do not change from t^n to t^{n+1}, the integral term over the set of boundary faces with Dirichlet boundary conditions, $\int_{\mathcal{F}_D} C_{11}v^{n+1}\phi_h^n dA$, can always be evaluated as presented in Section 3.3.

This implies that we only have to show how the cut-cell integral terms of $b(v^{n+1},\mathbf{D}_h^n)$ and $s(v^{n+1},\phi_h^n)$ in (3.181) are evaluated. As shown in (3.169)-(3.172) the shape functions in cut-cells within the cut-cell domain are equivalent to the shape functions v and $\boldsymbol{\tau}$ which are defined on the hexahedral element that contains the cut-cell. Furthermore, the shape functions vanish outside of the cut-cell domain. Since the cut-cell domains change from t^n to t^{n+1}, the support of the shape functions also changes. This implies that the evaluation of the surface integral terms $b(v^{n+1},\mathbf{D}_h^n)$ and $s(v^{n+1},\phi_h^n)$ on cut-cells faces in (3.181), which are given by

$$\int_{\mathcal{F}_0}\{v^{n+1}\}[\![\mathbf{D}_h^n]\!]dA+\int_{\mathcal{F}_0} C_{11}[\![\phi_h^n]\!]\cdot[\![v^{n+1}]\!]dA, \tag{3.184}$$

[6] The parameter $\mathbf{C}_{12}$ is set to $\mathbf{C}_{12}=\mathbf{0}$ in the bilinear form of the EBCDG method. As shown in Section 3.2.6, this parameter value also satisfies all of the desired properties.

becomes extremely difficult. Furthermore, accounting for changes of cut-cell domains, the change in shape functions on each face results in high additional computational costs.
By contrast, the volume term of $b(v^{n+1}, \mathbf{D}_h^n)$ in (3.181) which is given by

$$-\sum_{K \in \Omega_h^{n+1}} \int_K \nabla \cdot \mathbf{D}_h^n v^{n+1} d\mathbf{x}, \tag{3.185}$$

can be evaluated without huge additional costs, although the test functions and the approximate solutions belong to different finite element spaces. For this reason, the idea is to transform surface integral terms $b(v^{n+1}, \mathbf{D}_h^n)$ and $s(v^{n+1}, \phi_h^n)$ on cut-cells faces in (3.181) to volume integrals. In the next section, we present this transformation in detail.

3.6.2.1 Transformation of face integral terms to volume integral terms

In this section, we show that the surface integral terms $b(v^{n+1}, \mathbf{D}_h^n)$ and $s(v^{n+1}, \phi_h^n)$ in (3.181) can be expressed by volume integral terms. The surface integral terms $b(v^{n+1}, \mathbf{D}_h^n)$ and $s(v^{n+1}, \phi_h^n)$ can be transformed to volume integrals by using the lifting operators $\mathcal{M} : L^2(\mathcal{F}) \rightarrow V_h$ and $\mathcal{L} : [L^2(\mathcal{F})]^3 \rightarrow V_h$ as shown in [71] and [7]. These lifting operators are given by

$$\int_{\Omega_h} \mathcal{M}(\llbracket \mathbf{q} \rrbracket) \phi \, dV = \int_{\mathcal{F}} \llbracket \mathbf{q} \rrbracket \{\phi\} dA \quad \text{and} \quad \int_{\Omega_h} \mathcal{L}(\llbracket v \rrbracket) \phi \, d\mathbf{x} = \int_{\mathcal{F}} \llbracket v \rrbracket \cdot \llbracket \phi \rrbracket dA. \tag{3.186}$$

Applying these lifting operators to the integral terms in (3.181) yields

$$\begin{aligned} &- \sum_{K_i^{n+1} \in \Omega_h^{C,n+1}} \int_{K_i^{n+1}} \nabla \cdot \mathbf{D}_h^n v^{n+1} d\mathbf{x} + \int_{\mathcal{F}_0} \{v^{n+1}\} \llbracket \mathbf{D}_h^n \rrbracket dA + \int_{\mathcal{F}_0} C_{11} \llbracket \phi_h^n \rrbracket \cdot \llbracket v^{n+1} \rrbracket dA \\ &+ \int_{\mathcal{F}_D} C_{11} v^{n+1} \phi_h^n dA + g(v^{n+1}) \\ &= \sum_{K_i^{n+1} \in \Omega_h^{C,n+1}} \int_{K_i^{n+1}} \left(-\nabla \cdot \mathbf{D}_h^n + \mathcal{M}(\llbracket \mathbf{D}_h^n \rrbracket) + C_{11} \mathcal{L}(\llbracket \phi_h^n \rrbracket) \right) v^{n+1} dV. \end{aligned} \tag{3.187}$$

In the following, the integrand term of the volume integral in (3.188) is denoted as

$$\rho_h^n := -\nabla \cdot \mathbf{D}_h^n + \mathcal{M}(\llbracket \mathbf{D}_h^n \rrbracket) + C_{11} \mathcal{L}(\llbracket \phi_h^n \rrbracket), \tag{3.188}$$

since it can be interpreted as an approximation of the charge density ρ which is defined as $\rho = \nabla \cdot \mathbf{D}$. Furthermore, the term ρ_h^n equates to the analytical charge density ρ, if the exact potential and electric field solution ϕ and $\mathbf{D}$ of the field problem are inserted in 3.188. This is because the jump and the average of the exact solutions vanish at the interface, i.e. $\llbracket \mathbf{D}^n \rrbracket = 0$ and $\llbracket \phi_h^n \rrbracket = \mathbf{0}$.
Therefore, the integral terms in (3.187) are written in the form

$$\sum_{K_i^{n+1} \in \Omega_h^{C,n+1}} \int_{K_i^{n+1}} \rho_h^n v^{n+1} dV = -b\left(v^{n+1}, \mathbf{D}_h^n\right) + s\left(v^{n+1}, \phi_h^n\right) + g\left(v^{n+1}\right). \tag{3.189}$$

However, the lifting operators are not evaluated in this work. We just use this theoretical result to justify the approximation of the integral terms in (3.181) in cut-cells which is presented in the next section.

3.6.2.2 Approximation of the volume integral terms

In this section, we show how the volume integral term in (3.189) can be used to obtain an approximation of (3.181). The idea is to evaluate an approximation of the above introduced charge density ρ_h^n. Then, the charge density ρ_h^n is entered into the volume integral term on the left hand side of (3.189) to obtain an approximation of (3.181). The evaluation of the left hand side term of (3.189) is discussed in section 3.6.2.3.

Note that the transformation presented in Section 3.6.2.1 can also be applied to the following term:

$$-b\left(v^n, \mathbf{D}_h^n\right) + s\left(v^n, \phi_h^n\right) + g\left(v^n\right). \tag{3.190}$$

The charge density ρ_h^n satisfies

$$\sum_{K_i^n \in \Omega_h^{C,n}} \int_{K_i^n} \rho_h^n v^n dV = -b\left(v^n, \mathbf{D}_h^n\right) + s\left(v^n, \phi_h^n\right) + g\left(v^n\right). \tag{3.191}$$

The right-hand side of (3.191) can be evaluated as shown in Section 3.3 since the test and approximation functions in 3.191 are defined at the same point in time t^n and, therefore, belong to the same finite element spaces V_h^n and Σ_h^n. Therefore, equation (3.191) can be used to evaluate an approximation of ρ_h^n.

As mentioned before, the finite element spaces of the approximate solutions, $\mathbf{D}_h$ and ϕ_h, consist of the local finite element spaces $\mathcal{S}$ and $\mathcal{U}$, respectively, which are polynomial spaces. Therefore, the charge density ρ_h^n can also be approximated by polynomial local finite element spaces. Hence, ρ_h^n is approximated using an expansion of ρ_h^n in V_h^n which is given by[7]

$$\rho_h^n(\mathbf{x}) = \bigoplus_{K_i^n \in \Omega_h^{C,n}} \rho_{h|K_i^n}^n(\mathbf{x}), \tag{3.192}$$

$$\rho_{h|K_i^n}^n(\mathbf{x}) \approx \sum_{j=1}^{N^p} \rho_{h,j}^{K_i^n,n} v_j^n(\mathbf{x}), \tag{3.193}$$

where $v_j^n \in V_h^n$. Inserting (3.192) in (3.191) yields

$$\sum_{K_i^n \in \Omega_h^{C,n}} \sum_{j=1}^{N^p} \int_{K_i^n} \rho_{h,j}^{K_i^n,n} v_j^n(\mathbf{x}) v^n dV \approx -b\left(v^n, \mathbf{D}_h^n\right) + s\left(v^n, \phi_h^n\right) + g\left(v^n\right). \tag{3.194}$$

[7] Note that for our choice of H^1 and $H(div)$-conforming shape functions the local finite element spaces $\mathcal{S}$ and $\mathcal{U}$ span the polynomial spaces $Q^{p,p,p}$ and $Q^{p+1,p,p} \times Q^{p,p+1,p} \times Q^{p,p,p+1}$ (see, e.g Section 3.2.7). Therefore, the expansion of ρ_h^n in terms of basis functions $v^n \in \mathcal{S}^n$ is an approximation, since $\mathcal{M}(\llbracket \mathbf{D}_h^n \rrbracket) \in \Sigma_h$ and the local finite element spaces of Σ_h span the space $Q^{p+1,p,p} \times Q^{p+1,p,p} \times Q^{p+1,p,p}$. For a finite element space $\mathcal{U}(K)$ which spans the polynomial space $[Q^{p,p,p}]^3$, the term ρ_h could be exactly expanded in terms of shape functions $v^n \in V_h$, i.e. $\rho_h^n = \sum_{i=1}^{Np} \rho_{h,k}^n v^n$. Nonetheless, the numerical results presented in Section 3.6 demonstrate that the approximation in this case yields accurate results. Furthermore, the convergence study in 4.2 shows convergence of the method with respect to the energy and the dissipation of the system (see Section 4.2).

Since the right-hand side term of (3.194) can be evaluated as shown in Section 3.3, an approximation of ρ_h^n is obtained by solving (3.194) for the coefficients $\rho_{h,j}^{K_i^n,n}$. Note that the direct solver introduced in Section 3.5 is used to solve (3.194) for the coefficients $\rho_{h,j}^{K_i^n,n}$.
Then, the approximation of ρ_h^n is entered in (3.189) to evaluate an approximation of (3.181). Entering ρ_h^n in (3.189) yields

$$\sum_{K_i^{n+1}\in\Omega_h^{C,n+1}} \int_{K_i^{n+1}} (\sum_{j=1}^{N^p} \rho_{h,j}^{K_i^n,n} v_j^n) v^{n+1} dV \approx \sum_{K_i^{n+1}\in\Omega_h^{C,n+1}} \int_{K_i^{n+1}} \rho_h^n v^{n+1} dV = \tag{3.195}$$

$$-b\left(v^{n+1}, \mathbf{D}_h^n\right) + s\left(v^{n+1}, \phi_h^n\right) + g\left(v^{n+1}\right). \tag{3.196}$$

Note that the integral term

$$\sum_{K_i^{n+1}\in\Omega_h^{C,n+1}} \int_{K_i^{n+1}} (\sum_{j=1}^{N^p} \rho_{h,j}^{K_i^n,n} v_j^n) v^{n+1} dV \tag{3.197}$$

in (3.195) still contains shape functions which are defined at t^n and t^{n+1} and, therefore, belong to the different finite element spaces (V_h^n, Σ_h^n) and (V_h^{n+1}, Σ_h^{n+1}). As a next step, we will show how the term (3.197) is evaluated with respect to the shape functions v^n and v^{n+1}. For the sake of simplicity, we use the matrix formulation of the term (3.197) which is given by

$$\mathbf{P}^{n,n+1}\boldsymbol{\rho} \quad \text{with} \quad \left(\mathbf{P}^{n,n+1}\right)_{ij} = \sum_{K_i^{n+1}\in\Omega_h^{C,n+1}} \int_{K_i^{n+1}} v_j^n v_i^{n+1} dV, \tag{3.198}$$

where $\boldsymbol{\rho}$ denotes the vector of the coefficients $\rho_{h,j}^{K_i^n,n}$.

3.6.2.3 Evaluation of the components of matrix $P^{n,n+1}$

In this section, we present how the matrix components of $\mathbf{P}^{n,n+1}$ are evaluated. The matrix components of $\mathbf{P}^{n,n+1}$ on cut-cells are evaluated by exploiting the fact that the cut-cell shape functions, v^n and v^{n+1}, are equivalent to the shape function v within the cut-cell domain. As shown in (3.169), this shape function is defined on the hexahedral element in which the cut-cell is located. Figure 3.8 illustrates two exemplary cut-cells, K_C and K_D, which change their volume from time t^n to t^{n+1}. The cut-cell domain of K_C grows from t^n to t^{n+1}, whereas the cut-cell domain of K_D shrinks form from t^n to t^{n+1}. By definition, the shape functions $v^{C,n}$ and $v^{C,n+1}$ of K_C^n and K_C^{n+1}, respectively, are equivalent on the cut-cell volume of K_C^n. However, $v^{C,n}$ vanishes outside of K_C^n. This implies that the matrix coefficients of $\mathbf{P}^{n,n+1}$ for cut-cell K_C^{n+1} satisfy

$$\left(\mathbf{P}^{n,n+1}|_{K_C^{n+1}}\right)_{ij} = \int_{K_C^{n+1}} v_j^{C,n} v_i^{C,n+1} dV = \int_{K_C^n} v_j^{C,n} v_i^{C,n} dV, \tag{3.199}$$

since $v^{C,n}$ vanishes $\forall \mathbf{x} \in K_C^{n+1} \backslash K_C^n$ and $v^{C,n+1} = v^{C,n}$ $\forall \mathbf{x} \in K_C^n$. Therefore, the matrix coefficients of $\mathbf{P}^{n,n+1}$ of cut-cell K_C are evaluated using the right-hand side term of (3.199). The matrix

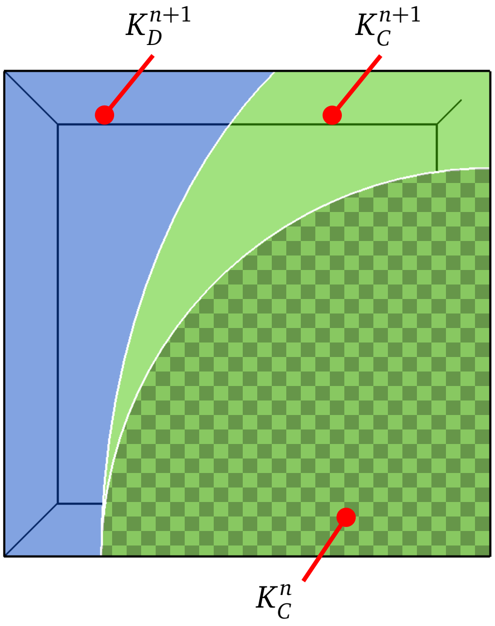

Figure 3.8.: Figure of a hexahedral cell containing two cut-cells which change their material domain from time t^n to t^{n+1}.

coefficients of $\mathbf{P}^{n,n+1}$ of cut-cell K_D are evaluated in a similar fashion since $\upsilon^{D,n+1}$ vanishes for all $\mathbf{x} \in K_D^n \setminus K_D^{n+1}$ and it further holds $\upsilon^{D,n} = \upsilon^{D,n+1}$ $\forall \mathbf{x} \in K_D^{n+1}$. This implies that the matrix coefficients of $\mathbf{P}^{n,n+1}$ of cut-cell K_D^{n+1} satisfy

$$\left(\mathbf{P}^{n,n+1}|_{K_D^{n+1}}\right)_{ij} = \int_{K_D^{n+1}} \upsilon_j^{D,n} \upsilon_i^{D,n+1} dV = \int_{K_D^{n+1}} \upsilon_j^{D,n+1} \upsilon_i^{D,n+1} dV. \tag{3.200}$$

As before, the matrix coefficients of $\mathbf{P}^{n,n+1}$ of cut-cell K_D^{n+1} are evaluated using the right-hand side term of (3.200).

Furthermore, we obtain additional non-zero matrix entries in $\mathbf{P}^{n,n+1}$ for cut-cell K_C^{n+1} since the shape functions $\upsilon^{D,n}$ of cut-cell K_D^n are not equal to zero on the domain $K_C^{n+1} \cap K_D^n = K_C^{n+1} \setminus K_C^n$. In this case, the matrix components are given by:

$$\left(\mathbf{P}^{n,n+1}|_{(K_C^{n+1} \cap K_D^n)}\right)_{ij} = \int_{K_C^{n+1} \cap K_D^n} \upsilon_j^{D,n} \upsilon_i^{C,n+1} dV = \int_{K_C^{n+1} \setminus K_C^n} \upsilon_j^{C,n+1} \upsilon_i^{C,n+1} dV. \tag{3.201}$$

The shape functions $\upsilon^{D,n}$ of cut-cell K_D^n on the domain $K_C^{n+1} \cap K_D^n = K_C^{n+1} \setminus K_C^n$ are equivalent to the shape functions $\upsilon^{C,n+1}$ on $K_C^{n+1} \cap K_D^n = K_C^{n+1} \setminus K_C^n$. Therefore, the matrix components of (3.201) are evaluated using the right-hand side term of (3.201). Note that the calculation of the matrix coefficients in (3.199), (3.200) and (3.201) must take into account the correct approximation orders p of the shape functions.

In the case of a shrinking cut-cell domain K_C and a growing cut-cell domain K_D from time t^n to t^{n+1}, the evaluation of the matrix coefficients of $\mathbf{P}^{n,n+1}$ is performed in a similar fashion. We only have to interchange the cut-cell domains in the (3.199), (3.200) and (3.201).

Moreover, when a normal element K_i at time t^n splits into two cut-cells K_C and K_D at t^{n+1}, the matrix coefficients are calculated in the following way:

$$\left(\mathbf{P}^{n,n+1}|_{K_C^{n+1}}\right)_{ij} = \int_{K_C^{n+1}} v_j^n v_i^{C,n+1} dV = \int_{K_C^{n+1}} v_j^{C,n+1} v_i^{C,n+1} dV, \tag{3.202}$$

$$\left(\mathbf{P}^{n,n+1}|_{K_D^{n+1}}\right)_{ij} = \int_{K_D^{n+1}} v_j^n v_i^{D,n+1} dV = \int_{K_D^{n+1}} v_j^{D,n+1} v_i^{D,n+1} dV. \tag{3.203}$$

In the reverse case, when cut-cells K_C^n and K_D^n become a normal element K_i at time t^{n+1}, the calculation is performed in a similar fashion.
Note that we have now presented the evaluation of the matrix components of $\mathbf{P}^{n,n+1}$. In a final step, we show how we treat normal elements which stay normal elements from t^n to t^{n+1} in matrix $\mathbf{P}^{n,n+1}$. As already mentioned, the shape functions in normal elements do not change at each point in time which means that $v^{n+1} = v^n = v$. This implies that the right-hand side term in (3.191) and (3.180) are equivalent. The right-hand side term in (3.191) and (3.180) on normal elements can be written in the form:

$$-b\left(v, \mathbf{D}_h^n\right) + s\left(v, \phi_h^n\right) + g\left(v\right). \tag{3.204}$$

Note that we consider the normal elements in the approximation of the charge density. However, we set the matrix entries of normal elements in $\mathbf{P}^{n,n+1}$ and $\mathbf{P}^{n,n}$, given by

$$\mathbf{P}^{n,n+1} = \mathbf{P}^{n,n} \quad \text{with} \quad (\mathbf{P}^{n,n})_{ij} = \sum_{K_i^n \in \Omega_h^{C,n}} \int_{K_i^n} v_i v_j dV, \tag{3.205}$$

to the identity matrix. This procedure doesn't change the right-hand side term in (3.206).
Having evaluated the matrix components of $\mathbf{P}^{n,n+1}$, the matrix is multiplied by the vector ρ of the coefficients. This yields the desired approximation term in (3.180).

3.6.3 Alternative approaches for the evaluation of the bilinear forms

In this section, we present two alternative approaches to evaluate the integral terms of $b(v^{n+1}, \mathbf{D}_h^n)$ and $s(v^{n+1}, \phi_h^n)$ in (3.181) which we have implemented and tested.
The first approach evaluates the integral terms of (3.181) by using the approximate solutions $\bar{\mathbf{D}}_h^n$ and $\bar{\phi}_h^n$ of $\mathbf{D}_h^n$ and ϕ_h^n which are projected to the cut-cell discretization at time t^{n+1} using a standard L^2-projection. More precisely, the L^2-projection of the approximate potential solution ϕ_h^n reads

$$\sum_{K_i^{n+1} \in \Omega_h^{C,n+1}} \int_{K_i^{n+1}} \left(\sum_{j=1}^{N^P} \bar{\phi}_{h,j}^{K_i^n,n} v_j^{n+1}\right) v_i^{n+1} dV = \sum_{K_i^{n+1} \in \Omega_h^{C,n+1}} \int_{K_i^{n+1}} \phi_h^n v_i^{n+1} dV, \tag{3.206}$$

where $\bar{\phi}_h^n = \sum_{j=1}^{N^P} \bar{\phi}_{h,j}^{k,n} v_j^{n+1}$ represent the approximate potential solution ϕ_h^n projected to the cut-cell discretization $\Omega_h^{C,n+1}$ at time t^{n+1}. The L^2-projection of the flux density term is conducted similarly. The approximation of the integral terms of (3.181) is then obtained by entering both projections of the approximate solutions in (3.181). Solving (3.179)-(3.180) for $\mathbf{D}_h^{n+1}$ and ϕ_h^{n+1} yields the approximate solutions at each point in time.

The second alternative approach is based on an approximation of the integral terms of $b(v^{n+1}, \mathbf{D}_h^n)$ and $s(v^{n+1}, \phi_h^n)$ in (3.181) which is given by:

$$- \sum_{K \in \Omega_h^{n+1}} \int_K \nabla \cdot \mathbf{D}_h^n v^{n+1} d\mathbf{x} + \int_{\mathcal{F}_D} C_{11} v^{n+1} \phi_h^n dA + g\left(v^{n+1}\right) \approx \\ -b\left(v^{n+1}, \mathbf{D}_h^n\right) + s\left(v^{n+1}, \phi_h^n\right) + g\left(v^{n+1}\right). \tag{3.207}$$

This approximation is motivated by the fact that the first integral term can be interpreted as an approximation of the charge density term since $\rho = \nabla \cdot \mathbf{D}$. Furthermore, the two interface terms vanish for the exact solutions ϕ and $\mathbf{D}$, i.e. $[\![\mathbf{D}^n]\!] = 0$ and $[\![\phi_h^n]\!] = \mathbf{0}$. Both surface integral terms of $b(v^{n+1}, \mathbf{D}_h^n)$ and $s(v^{n+1}, \phi_h^n)$ in (3.181) are considered as residual terms and, therefore, neglected in (3.207).

However, for both alternative approaches we didn't obtain numerically stable results. Only the approach presented in Section 3.6.2 yields numerically stable results. Furthermore, the EBCDG method combined with the approach presented in Section 3.6.2 shows convergence with respect to the energy and power dissipation for the chosen application example. The reader is referred to Chapter 4 for further details on the numerical results of the EBCDG method.

4 Numerical Examples

This chapter presents various numerical simulation results of the BCDG method introduced in Chapter 3. The outline of this chapter is as follows: First, we present the simulation results of the BCDG method on a verification example. We use a standard cylindrical capacitor field problem with two dielectric layers to assess the accuracy of the BCDG method. Several convergence studies and a comparison study using the commercial software CST STUDIO SUITE®2012 [1] demonstrate the high accuracy of the BCDG method for this example. Furthermore, we use this example to show that both the adaptive approximation order method and the cell merging method reduce the condition number of the linear system of equations substantially. Next, the simulation results of the BCDG method with and without the additional DG hybridization on the cylindrical capacitor example illustrate that this hybridization reduces the DG DOF significantly without affecting the high accuracy of the BCDG method. In Section 4.2, we present the numerical results of the EBCDG method for the application example of an oscillating water droplet lying on the insulation layer of a high voltage insulator. In contrast to the cylindrical capacitor example, this application example comprises a material domain that moves over time. The potential and electric field simulation results of the EBCDG method are shown at different points in time. Furthermore, a convergence study is presented which demonstrates the convergence of the EBCDG method with respect to the energy and power dissipation of the field problem.

4.1 Verification of the BCDG method using a cylindrical capacitor example

The BCDG method presented in Chapter 3 can be applied to both the EQS and the ES approximation of the Maxwell's equations which are introduced in Chapter 2. To assess the accuracy of the BCDG method on ES and EQS field problems with non-moving material domains, we apply the method to a verification example [63]. In this work, we choose the field problem of a cylindrical capacitor example filled with two dielectric layers (see Figure 4.1). Note that in this section we focus on field problems with non-moving material domains. Field problems which comprise moving materials are considered in Section 4.2.2. The cylindrical capacitor example represents an EQS field problem if time varying voltage excitations at low frequencies are applied to the cylindrical electrodes [90]. For constant voltage excitations the field distribution can be described by the ES approximation of the Maxwell's equations. In both cases the analytical solution is well-known. In various convergence studies in Section 4.1.3 the analytical solution is used to assess the accuracy of the numerical solutions of the BCDG method. Furthermore, in Section 4.1.4 the verification example is used to compare the simulation results of the BCDG method to the simulation results obtained with the commercial software CST STUDIO SUITE®2012 [1]. In Section 4.1.1, the modelling details of the verification example are presented.

4.1.1 Modelling of the cylindrical capacitor

The geometric outline of the numerical model of the cylindrical capacitor is shown in Figure 4.1. Since cylindrical capacitors are axially symmetric, it is sufficient to model only a quarter part of the cylindrical capacitor. For this reason, Figure 4.1 only shows a quarter part of the cylindrical capacitor. The blue and green material domains represent the dielectric materials layers which fill the space between the cylindrical electrodes. The green material layer is indicated by Ω_1. It is assumed to have length $l = 10mm$, radius $r_1 = 4mm$ and the material parameters ϵ_1 and κ_1. The blue material domain denoted as Ω_2 has the same length and radius, i.e. $l = 10mm$, $r_2 = 4mm$, but different material parameters ϵ_2 and κ_2. The values of the material parameters are specified below. Note that the real length L of the cylindrical capacitor example is assumed to be much longer than both length l and radius $r = r_1 + r_2$, i.e. $L >> (r_1 + r_2)$. We make this assumption so that the electric field in x-direction can be neglected. However, the computational domain doesn't need to be of length L. As the capacitor is axially symmetric, it is sufficient to model only a part of the cylindrical capacitor with length l and enforce the negligible electric field in x-direction by homogeneous Neumann boundary conditions.

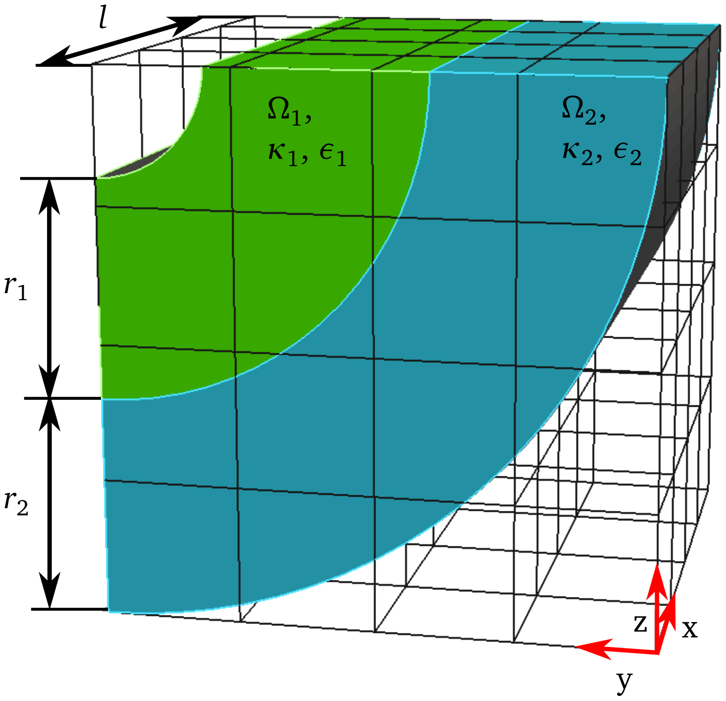

Figure 4.1.: Computational domain of the cylindrical capacitor verification example.

In Figure 4.1, the cylindrical electrodes are represented by the grey cylindrical sheets at the inner and outer cylindrical face of the blue and green material domains. Since the electrodes consist of perfectly conducting materials, we model the voltage excitation applied to the cylindrical electrodes as boundary conditions. This implies that the computational domain Ω of the cylindrical capacitor is given by $\Omega = \Omega_1 \cup \Omega_2$. The boundary $\Gamma = \partial\Omega$ of the computational domain is partitioned into boundaries with Dirichlet and Neumann boundary conditions. The Dirichlet boundary conditions enforce the voltage excitation which is applied to the cylindrical electrodes. In the EQS case we apply a sine voltage at a frequency of 50 Hz. Therefore, the values of the potential at the inner and outer cylindrical face are set to $\phi_{D1}(t) = \sin(2\Pi f t)V$

with $f = 50Hz$ and $\phi_{D2}(t) = 0V$. The initial value is $\phi_h(t_0 = 0) = \mathbf{0}$. In the ES case a constant voltage is enforced with $\phi_{D1} = 1V$ and $\phi_{D2}(t) = 0V$. Furthermore, we apply homogeneous Neumann boundary conditions to the remaining plane boundary faces, since the analytical electric fields on these faces vanish in normal direction.

Figure 4.1 furthermore shows an exemplary mesh with hexahedral elements illustrated by the black lines. This mesh completely covers the computational domain. In this example, we restrict ourselves to hexahedral elements with an equidistant element length h with $h \in [l/3, l/30]$.

Next, the LDG flux parameters and the Θ parameter need to be specified. The LDG flux parameters $\mathbf{C}_{12}$ and C_{11} are set to $\mathbf{C}_{12} = \mathbf{0}$ and $C_{11} = \mathcal{O}(c/h)$ where $c \in [0, -1]$. Note that the simulation results of the BCDG method on the cylindrical capacitor example show that the same order of magnitude is obtained for all $c \in [0, -1]$. Furthermore, the semi-discrete weak DG formulation in the EQS case is solved using the Θ-method with $\Theta = 1/2$ (Crank-Nicolson method), since it is a second-order method for this value (see Section 3.4 for more information).

As mentioned before, the analytical potential and electric field solutions for the above specified cylindrical capacitor example are well-known (see, e.g. [90] for further information). Using the parameter values $r = \sqrt{(y-l)^2 + (z-l)^2}$, $r_A = 2$, $r_I = 6$, $r_E = 10$, $r_1 = r_I - r_A$, $r_2 = r_E - r_I$, $\epsilon_1 = 6\epsilon_0$, $\epsilon_2 = \epsilon_0$, $\kappa_1 = 10^{-8}$ and $\kappa_2 = 10^{-12}$, the analytical potential solution of the EQS cylindrical capacitor example is given by:

$$\begin{aligned} \phi_{an}(r) &= \cos(2f\Pi t)(0.0614293 - 0.0886238\log(r)) + \\ &\quad \sin(2f\Pi t)(1.13938 - 0.20108\log(r)) \quad \forall r \text{ with } r_A < r < r_I, \end{aligned} \tag{4.1}$$

$$\begin{aligned} \phi_{an}(r) &= \cos(2f\Pi t)(-0.438872 + 0.1905996\log(r)) + \\ &\quad \sin(2f\Pi t)(3.51181 - 1.525158\log(r)) \quad \forall r \text{ with } r_I < r < r_E. \end{aligned} \tag{4.2}$$

In the ES case, the analytical potential solution for the same set of parameter values reads

$$\phi_{an}(r) = 1.16648 - 0.240179\log(r) \quad \forall r \text{ with } r_A < r < r_I, \tag{4.3}$$

$$\phi_{an}(r) = 3.31819 - 1.44107\log(r) \quad \forall r \text{ with } r_I < r < r_B. \tag{4.4}$$

The analytical electric field solution can be derived by entering (4.1) and (4.2) in equation (3.33). This step is omitted for the sake of brevity.

Note that all parameter values used to model the cylindrical capacitor are chosen randomly. Since we consider this cylindrical capacitor example as an artificial verification example for the BCDG method, we have tested many different sets of parameters. As we obtained similar results with respect to the rates of convergence with all tested sets of parameters, we decided to chose randomly the particular set of parameters which we just presented.

Having completely described the modelling of the cylindrical capacitor example, we now present the simulation results obtained with the BCDG method.

4.1.2 Simulation results of the BCDG method

In this section, we show the potential distribution of the numerical solution on the ES cylindrical capacitor example. Figure 4.2 illustrates the potential distribution when the BCDG method using the *cell merging method* and the *DG hybridization* is applied to this example. The above presented parameter values, grid parameter $h = l/4$ and approximation order $p = 3$ were used in this case. Furthermore, Figure 4.2 demonstrates that the BCDG method provides a high resolution potential distribution, even when we use a coarse mesh with four hexahedral elements in each spatial direction and approximation order $p = 3$.

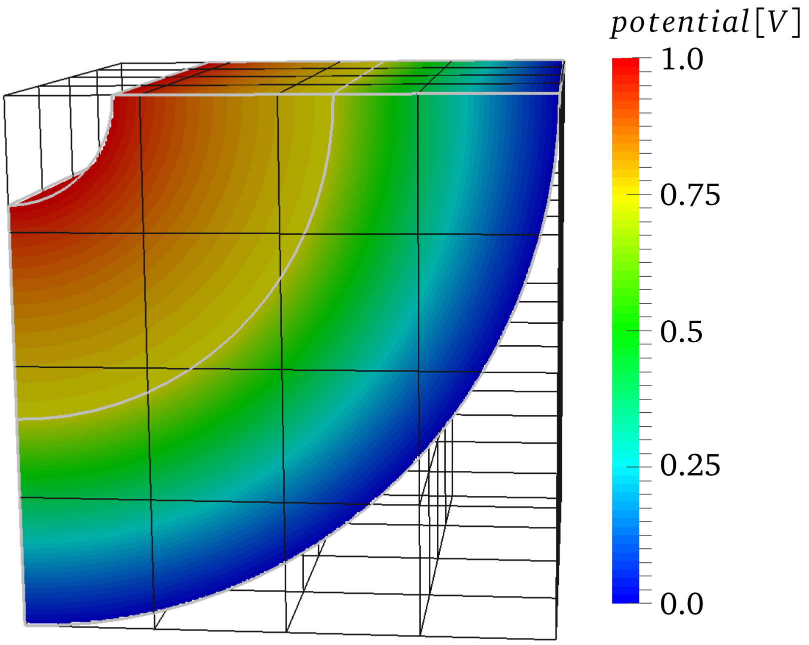

Figure 4.2.: Potential simulation results of the BCDG method with the *cell merging method* and the *DG hybridization* for the ES cylindrical capacitor example

4.1.3 Convergence study of the BCDG method using the analytical solution

In this section, we compare the numerical results of the BCDG method on the cylindrical capacitor example with the known analytical solution introduced in (4.1)-(4.4). Note that we present the convergence study for both the ES and the EQS cylindrical capacitor field problem. Since the BCDG method can be applied to the cylindrical capacitor example with either the *cell merging method* or the *adaptive approximation order method*, we show convergence studies for both supplementary methods. We first present the convergence study of the BCDG method combined with the *cell merging method*, before we show the convergence study of the BCDG method combined with the *adaptive approximation order method*. Note that both convergence studies are performed using the BCDG method with the additional *DG hybridization*.

To assess the accuracy of the simulation results, we require an appropriate accuracy measure. This accuracy measure is introduced in the next section.

4.1.3.1 Error criterion

In both convergence studies the accuracy of the BCDG method is presented with respect to the relative error $\varepsilon_{L^2}^{rel}$ measured in the broken L^2-norm, a commonly used accuracy measure for DG methods [38]. We need to distinguish between the relative error in the ES and EQS case. In the ES case, the relative error $\varepsilon_{L^2}^{rel}$ for a scalar function $u \in V_h^C$ and a vector function $\mathbf{v} \in \Sigma_h^C$ is defined by

$$\varepsilon_{L^2,\phi}^{rel} = \frac{\left\|\phi_h - \phi_{an}\right\|_{L^2}}{\left\|\phi_{an}\right\|_{L^2}} \qquad \text{and} \qquad \varepsilon_{L^2,\mathbf{E}}^{rel} = \frac{\left\|\mathbf{E}_h - \mathbf{E}_{an}\right\|_{L^2}}{\left\|\mathbf{E}_{an}\right\|_{L^2}}, \tag{4.5}$$

where the L^2-norm of the scalar and vector functions are given by:

$$\|u\|_{L^2}^2 = \sum_{K_i \in \Omega_h^C} \int_{K_i} u^2 dV \qquad \text{and} \qquad \|\mathbf{v}\|_{L^2}^2 = \sum_{K_i \in \Omega_h^C} \int_{K_i} \mathbf{v} \cdot \mathbf{v} dV, \tag{4.6}$$

respectively. In the EQS case, the relative error $\varepsilon_{L^2}^{rel}$ on a time interval $[t_a, t_e]$ with $t_a, t_e \in \mathbb{R}$ is defined as follows:

$$\varepsilon_{L^2,\phi}^{rel} = \frac{\max_{t \in [t_a,t_e]} \left\|\phi_h - \phi_{an}\right\|_{L^2}}{\max_{t \in [t_a,t_e]} \left\|\phi_{an}\right\|_{L^2}} \qquad \text{and} \qquad \varepsilon_{L^2,\mathbf{E}}^{rel} = \frac{\max_{t \in [t_a,t_e]} \left\|\mathbf{E}_h - \mathbf{E}_{an}\right\|_{L^2}}{\max_{t \in [t_a,t_e]} \left\|\mathbf{E}_{an}\right\|_{L^2}}, \tag{4.7}$$

where $t_e = t_a + 1/f$ and $T = t_e - t_a$. The time value T represents one period of the sinusoidal voltage excitation.

4.1.3.2 BCDG method with the cell merging method and the DG hybridization

The presentation of the convergence studies for the BCDG method with the *cell merging method* and the *DG hybridization* is organized as follows: First, the used time step, time interval and threshold parameter values of the method are discussed. Then, the relative error results for the EQS and the ES cylindrical capacitor field problem are shown.

As stated in Section 3.3.4.2, the *cell merging method* uses a threshold $\mathcal{T}_M = t_M V_n$, which depends on the volume V_n of the normal hexahedral elements in the mesh. The threshold value t_M is a dimensionless parameter which differs across approximation order p. We obtain the threshold values by performing calibration tests on the cylindrical capacitor example. The simulation results presented in Figure 4.3 are based on the following threshold values: $t_M = 0.001$ for $p = 1$, $t_M = 0.05$ for $p = 2$ and $t_M = 0.16$ for $p = 3$. A threshold value of $t_M = 0.16$ for $p = 3$ implies that all cut-cells with a volume smaller than 16% of the volume V_n of normal mesh elements are merged to neighbouring elements. Note that the threshold value increases with the approximation order. Since the increase in threshold value t_M from $p = 2$ to $p = 3$ is significant and, furthermore, the threshold value $t_M = 0.16$ for $p = 3$ is already high, we expect the threshold value t_M for $p = 4$ to be greater than 0.25. From the implementation point of view, a threshold value $t_M > 0.25$ is not feasible, since too many cut-cells would have to be merged. Therefore, we refrain from considering approximation orders with $p \geq 4$.

The convergence study presented below shows convergence with respect to the grid parameter h. For this convergence study we apply the Θ-method with the time step $\Delta t = 3.125\text{E-}6$. This time step is obtained from a further convergence study of the BCDG method with respect to the time step Δt. For this time step, the study shows that the error caused by the Θ-method can be neglected.

Note that the relative error of the EQS approximation is measured in the interval $[t_a = 0.08s, t_e = 0.1s]$. For the EQS cylindrical capacitor example, the initial conditions $\phi_h(t_0) = \mathbf{0}$ are used as they are straightforward to implement. However, these initial conditions differ to the assumed initial conditions of the homogeneous analytical solution presented in Section 4.1.1. This implies that we enter an additional error term in the calculation of the relative error if the analytical solution in (4.1)-(4.2) is used. To check at which time the additional error of the relative error term caused by the different initial conditions can be neglected, we use the relaxation time τ_e (see, e.g. [37]). It indicates how fast the additional error with respect to the different initial conditions decays to zero. For our example, we find that at $t_a = 0.08s$ the potential solution of the non-homogeneous potential initial conditions is of order $e^{-t_a/\tau_e} = e^{-t_a/(\epsilon/\kappa)} \approx 0$ and, therefore, negligible.

Figure 4.3 shows the relative errors of the EQS potential and electric field solution, ϕ_h and $\mathbf{E}_h$, of the BCDG method combined with the additional *cell merging method* and the *DG hybridization* for different grid parameters h and approximation orders p. Note that the grid parameter $h \in [1/30, 1/4]$ is the reciprocal of the number of elements along each axis. Since we use equidistant hexahedral mesh elements, the number of elements are equal for each axis. The relative error is evaluated for the approximation orders $p = 1, 2, 3$ of the shape functions (see Section 3.2.7 for more details). In Figure 4.3, the square symbols represent the relative errors of the potential ϕ_h and electric field $\mathbf{E}_h$ obtained for various grid parameters h and for each approximation order $p = 1, 2, 3$. The coloured lines represent fitted lines through the relative errors. Since we use a log-log scale the slope (gradient) of the lines indicates the rate of convergence of the relative error for the applied approximation order. The convergence of the relative error is approximately of order $O(h^R)$ where R denotes the slope of the line and, therefore, the rate of convergence. The obtained rates of convergence R are listed in the legend of Figure 4.3. The black lines in Figure 4.3 represent the reference rates of convergence $\mathcal{O}(h^{p+1})$ of the L^2 error estimate for the standard LDG method. The relative error of the BCDG method using the *cell merging method* converges approximately at the rate of convergence, $p+1$, for the electric potential and at rate, p, for the electric field for the approximation orders $p = 1, 2$ and $p = 3$. These results are in line with the L^2 error estimate for the standard LDG method on hexahedral elements presented in Section 3.2.6. This implies that the cut-cell discretization with the cell-

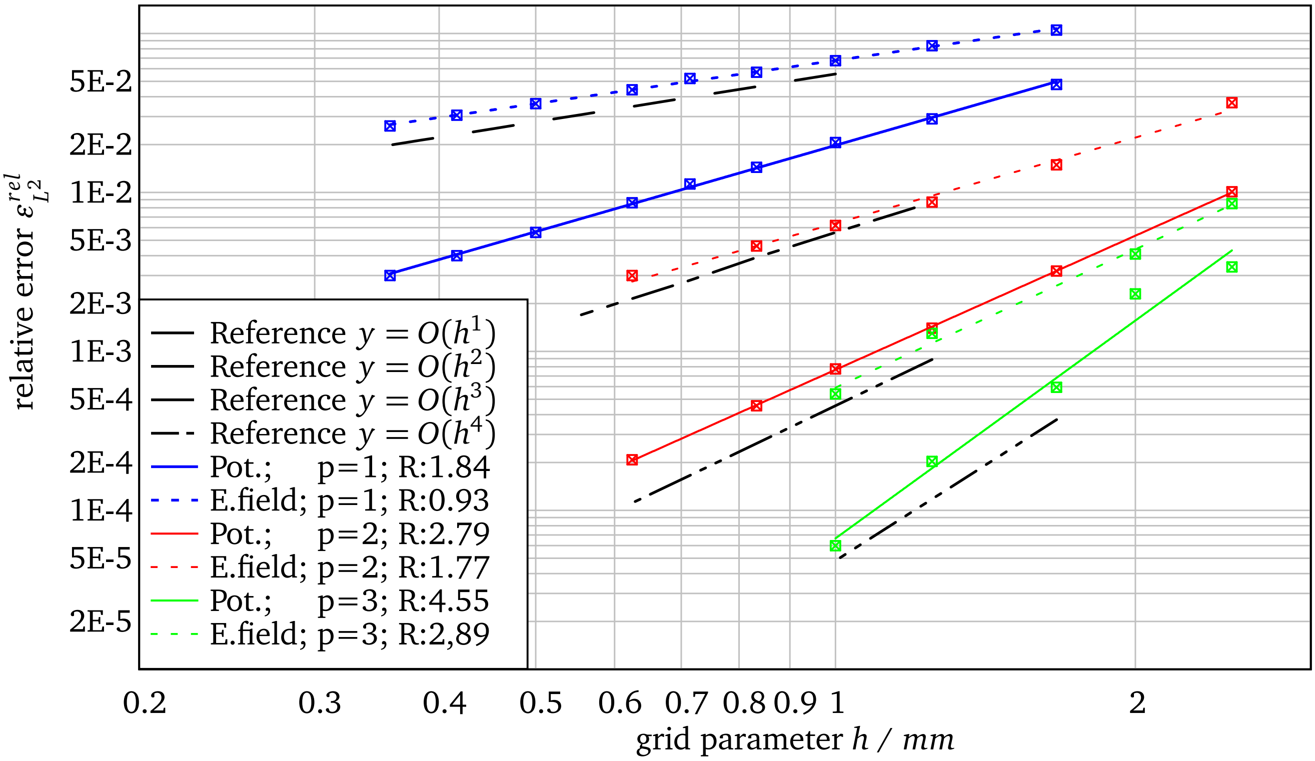

Figure 4.3.: L^2-convergence plot of the EQS potential and the electric field solutions of the BCDG method combined with the *cell merging method*.

merging method and the *DG hybridization* doesn't influence the acccuracy of the standard DG framework. Furthermore, our results are in accordance with the convergence rates reported by other authors using the standard LDG method on hexahedral elements (see, e.g. [20], [17] or [65]).

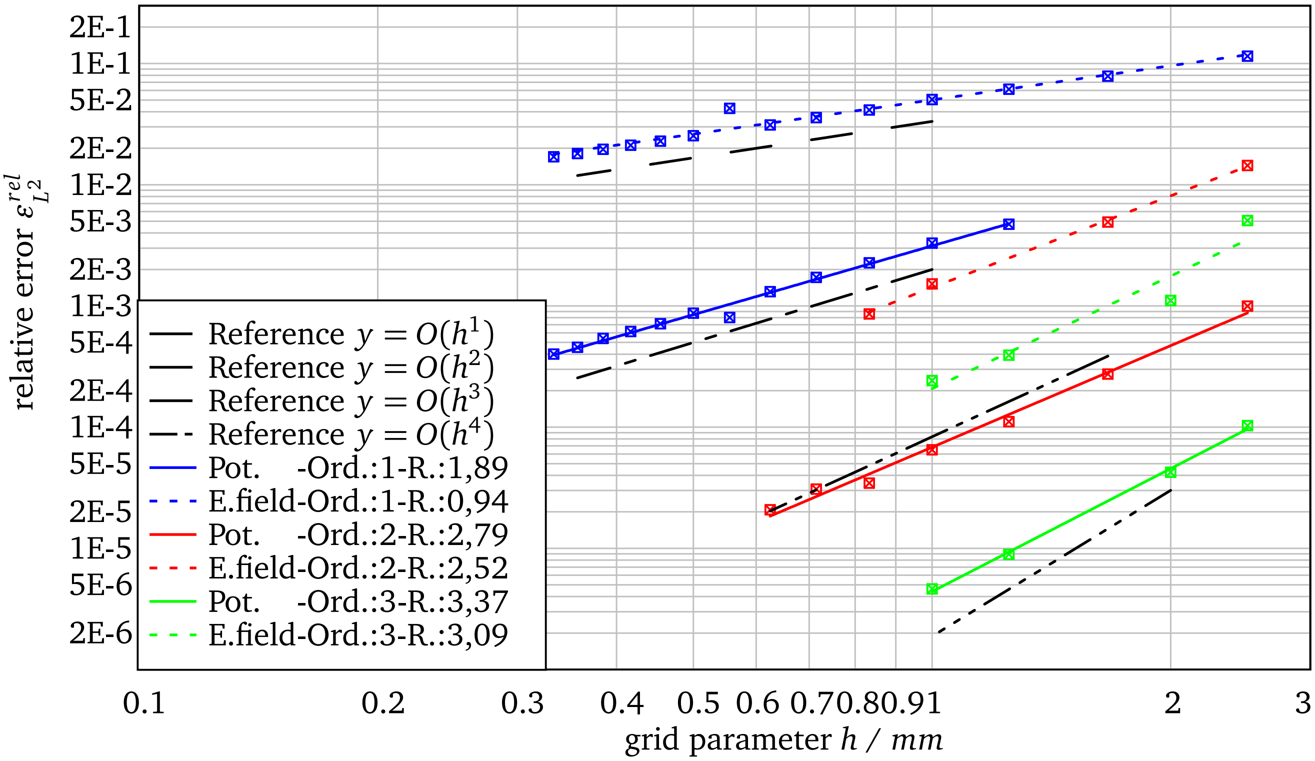

Figure 4.4.: L^2-convergence plot of the ES potential and the electric field solutions of the BCDG method combined with the *cell merging method*.

For the ES cylindrical capacitor field problem similar relative error results are obtained. Figure 4.4 shows the relative ES errors with respect to the grid parameter h for each approximation order $p = 1, 2, 3$. As before, the square symbols represent the relative errors obtained. The coloured lines are fitted lines through the relative ES errors of the BCDG method. The BCDG method using the *cell merging method* and the *DG hybridization* converges approximately with the rate of convergence $p + 1$ for the electric potential and p for the electric field for $p = 1, 2$ and $p = 3$ (see Section 3.2.6). Hence, in the ES case we obtain optimal convergence rates for the BCDG method, too. Finally, we would like to state that the rates of convergence we obtain are similar to the convergence results which have been reported by other authors using similar cut-cell discretization approaches on the Nitsche's method (see, e.g. [48], [57], [9]).

4.1.3.3 BCDG method with the adaptive approximation order method and the DG hybridization

In this section, we assess the accuracy of the BCDG method combined with the *adaptive approximation order method* and the *DG hybridization*. We apply the method again to the EQS and ES cylindrical capacitor field problem. The *adaptive approximation order method* divides the set of small cut-cells into different sets of small cut-cells according to their volume. Therefore, several thresholds are applied to the cut-cells in the mesh. As presented in Section 3.3.4.1 the thresholds of the *adaptive approximation order method* are denoted as $\mathcal{T}_i = t_i V_n$, $\forall p_i < p_p$ with $i = 1, ..., p-1$. The parameter t_i represents the threshold value which differs across approximation orders p. However, in our implementation for $p = 3$ we set t_1 equal to t_2. The quantity V_n represents the volume of the normal hexahedral elements of the mesh. The simulation of the cylindrical capacitor example is conducted using the following combination of threshold values for t_i: $t_0 = 0.001$ for $p \geq 1$; $t_1 = 0.05$ for $p = 2$ and $t_1 = t_2 = 0.16$ for $p = 3$. These values are determined by calibration tests on the cylindrical capacitor examples. As described in Section 3.3.4.1 the staircasing approximation is applied to cut-cells which have a volume smaller than the threshold $\mathcal{T}_0 = t_0 V_n$ for $p \geq 1$. Furthermore, the approximation order $p = 1$ is assigned to cut-cell K_i^C which belong to the set of cut-cells with volumes smaller than $\mathcal{T}_1 = t_1 V_n$ for $p = 2, 3$ (i.e. $K_i^C \in \Omega_h^{CCp_1}$, see Section 3.3.4.1 for more details). As before, we restrict ourselves to the approximation orders $p = 1, 2, 3$. Figure 4.5 shows the relative errors of the BCDG method for

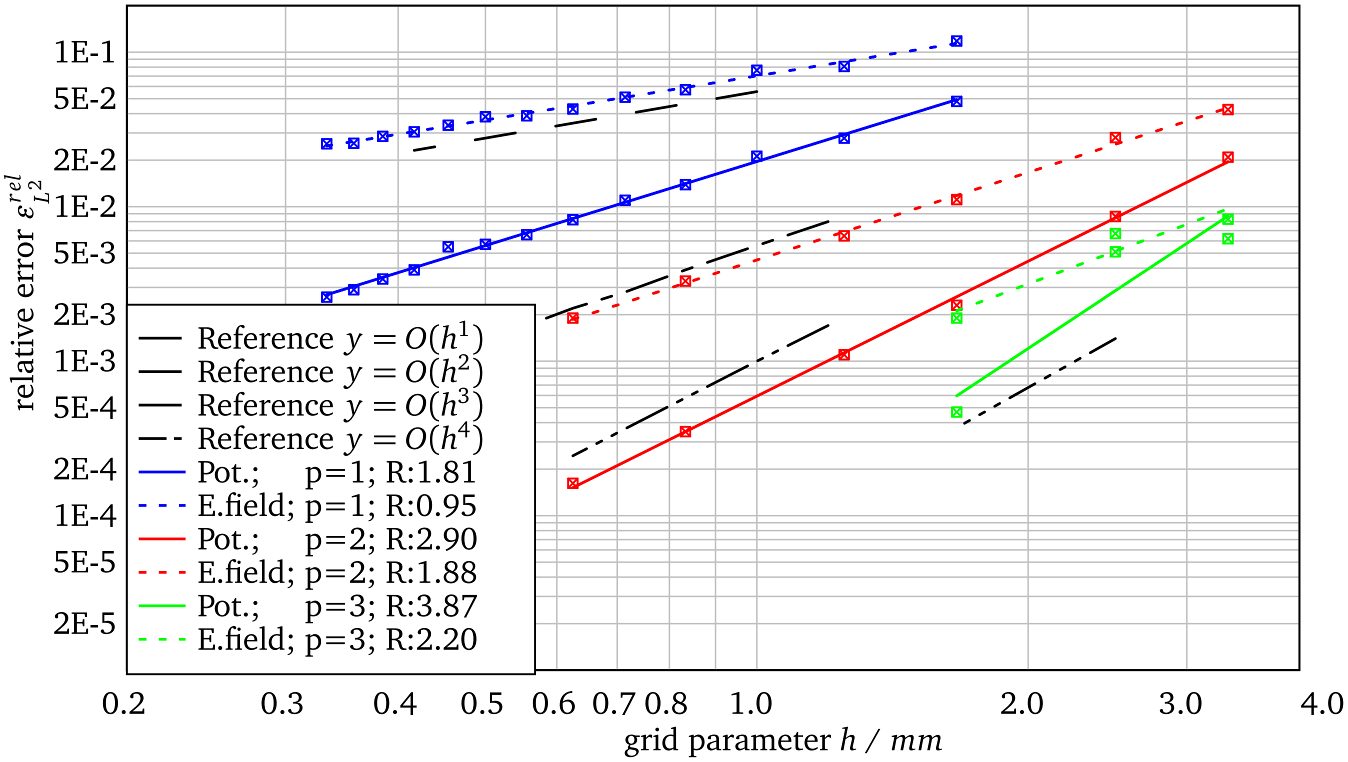

Figure 4.5.: L^2-convergence plot of the EQS potential and the electric field solutions of the BCDG method combined with the *adaptive approximation order method*.

the EQS cylindrical capacitor field problem. Both the relative electric potential errors and the electric field errors for the different grid parameters h and approximation orders p are shown. The square symbols represent the relative errors. The coloured fitted lines and the reference lines are specified in the same way as in Figure 4.3 and Figure 4.4. Figure 4.5 illustrates that the rates of convergence of the BCDG method using the *adaptive approximation order method* and the *DG hybridization* are approximately of order $p+1$ for the approximate potential solution and of order p for the approximate electric field solution for $p = 1, 2$.

The relative errors obtained for the ES cylindrical capacitor example are presented in Figure 4.6. We approximately obtain the optimal rates of convergence $p + 1$ for the ES potential solution and p for the ES electric field solution for $p = 1, 2$. However, for approximation order $p = 3$ the rate of convergence of the potential and the electric field solution of the ES and EQS cylindrical

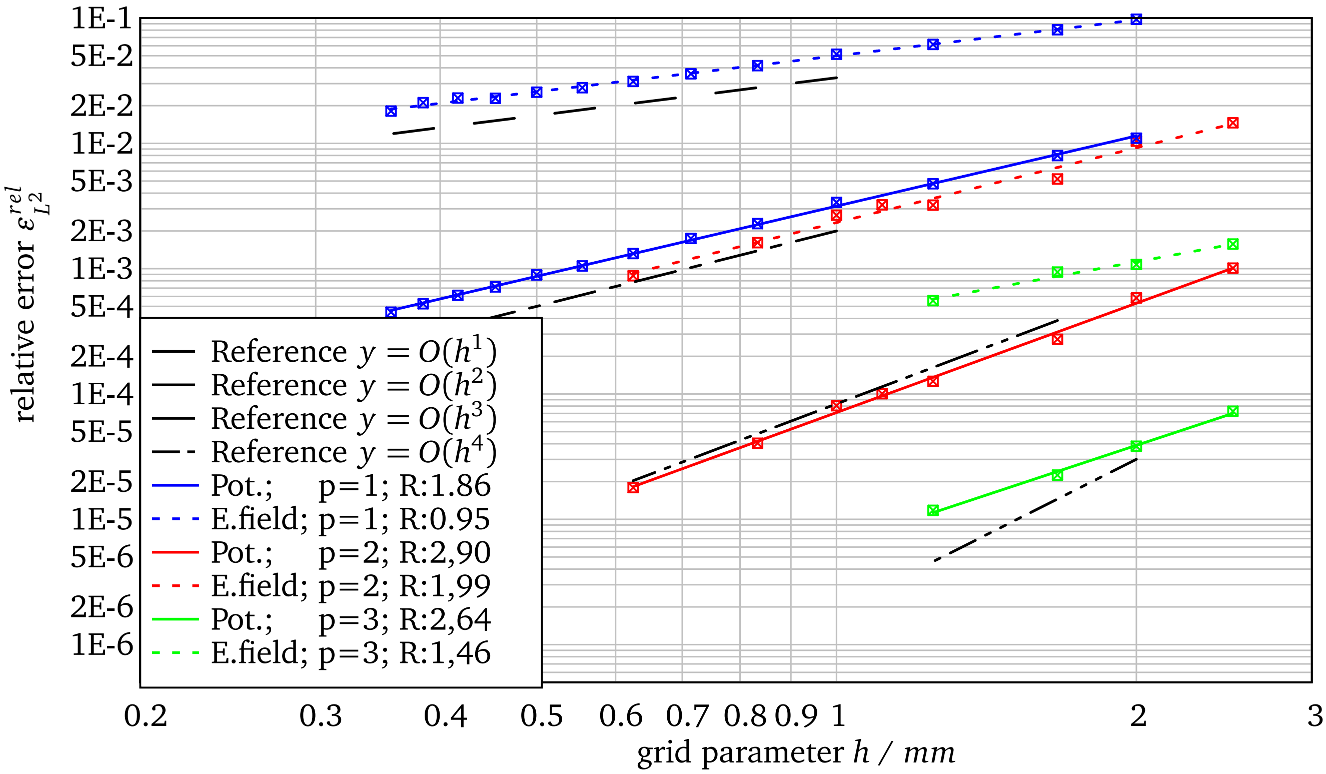

Figure 4.6.: L^2-convergence plot of the ES potential and the electric field solutions of the BCDG method combined with the *adaptive approximation order method*.

capacitor example is slightly lower than $p + 1$ and p, respectively. This might be caused by assigning the approximation order $p = 1$ to all cut-cells with a volume smaller than $\mathcal{T}_1 = 0.16V_n$. We believe that the rates of convergence for approximation order $p = 3$ become optimal if another threshold value $\mathcal{T}_2 > \mathcal{T}_1$ is introduced. With two threshold values, we could first lower the approximation order to $p = 2$, before we use $p = 1$ for very small cut-cells[1]. However, the trimming of the convergence rate to optimal orders by changing the threshold values is left for future work.

The relative error results shown in Figure 4.3-4.6 demonstrate that the BCDG method combined with either the *cell merging method* or the *adaptive approximation order method* yields optimal convergence rates for $p = 1, 2$. This implies that the accuracy of the DG framework in the EQS and ES case in not affected for $p = 1, 2$. For $p = 3$ the BCDG method combined with the *cell merging method* yields slightly better results than the BCDG method combined with the *adaptive approximation order method*. However, for $p = 3$ the slightly lower accuracy of the BCDG method combined with the *adaptive approximation order method* is outweighed by the method's simplicity which results in slightly lower computational costs. Since the differences in accuracy and computational costs are rather small, we conclude that both methods are well suited for ES and EQS field applications.

[1] For instance, the lower order $p_2 = 2$ could be assigned to cut-cell domains K_i^C which satisfy $0.05V_n \leq Volume(K_i^C) \leq 0.16V_n$

4.1.4 Comparison study with the commercial software CST STUDIO SUITE®2012

In this section we compare the simulation results of the BCDG method with simulation results obtained with the commercial software *CST STUDIO SUITE®2012*[1]. The simulation results which we present below are obtained by conducting simulations on the ES cylindrical capacitor field problem presented in Section 4.1.1. As CST EM STUDIO®2012 doesn't include a transient solver for EQS field problems, we restrict ourselves to the ES case [1]. Since the relative errors of the BCDG method for the EQS and ES cylindrical capacitor example presented in Section 4.1.3 approximately converge at the same rates of convergence, we consider it to be sufficient to only compare the simulation results obtained on the ES cylindrical capacitor example.

To simulate the ES cylindrical capacitor example with CST EM STUDIO®2012 we use the *Electrostatics Field Solver*. This solver is based on a finite element method. In order to exactly model the cylindrical capacitor, the simulations in *CST EM STUDIO®2012* are performed using a tetrahedral mesh with curved elements up to order three [1]. Note that we use the same CAD model of the cylindrical capacitor for the simulation with CST EM STUDIO®2012 as well as with the BCDG method. The boundary conditions are identically enforced in both methods. Furthermore, the following material parameter values are assumed: $\epsilon_1 = \epsilon_0$ and $\epsilon_2 = 6\epsilon_0$. All other parameter values[2] are set to the values introduced in Section 4.1.1. Furthermore, the simulations of the BCDG method on the cylindrical capacitor are performed with the threshold values introduced in Section 4.1.3.

In this section, the accuracy of the numerical solutions obtained with CST EM STUDIO®2012 and the BCDG method is presented using a relative error measure for the ES energy. This error measure is chosen because the ES energy can easily be evaluated in and exported from CST EM STUDIO®2012.

This relative error measure is calculated for the simulation results obtained with CST EM STUDIO®2012 and the BCDG method combined with both the *cell merging method* and the *adaptive approximation order method* with respect to the analytical solution. The relative error is defined as follows:

$$\varepsilon_E^{rel} = \frac{W_{e,h} - W_{e,an}}{W_{e,an}}, \tag{4.8}$$

with

$$W_{e,h} = \frac{1}{2}\int_\Omega \mathbf{E}_h \cdot \mathbf{D}_h dV \qquad \text{and} \qquad W_{e,an} = \frac{1}{2}\int_\Omega \mathbf{E}_{an} \cdot \mathbf{D}_{an} dV, \tag{4.9}$$

where $W_{e,h}$ denotes the ES energy of the simulation results and $W_{an,h}$ the analytical ES energy (see Section 2.2.2.3 for more information). Note that approximate electric field and flux density solution results of the cylindrical capacitor example obtained from CST EM STUDIO®2012 or the BCDG method are denoted as E_h and D_h in (4.9) . The quantities E_{an} and D_{an} in (4.9) represent the analytical electric field and flux density solutions of the cylindrical capacitor example.

[2] The permittivity values are changed because we obtain errors with respect to the energy of the BCDG method which are of order 3E-6 for $p = 3$ and the largest grid size $h = 10/4$. As presented in Section 3.3.3, we use the geometry kernel OCCT in our BCDG method to evaluate the integrals of the DG formulation. Since the precision value in OCCT is set to 1E-7, we are not able to show simulation results of the BCDG method which are lower than 1E-7. Therefore, we changed the permittivity values to obtain larger error values for $p = 3$ which enable us to show convergence rates also for $p = 3$.

The relative error results of the simulations on the ES cylindrical capacitor example are shown in Figure 4.7. Note that we show the relative errors with respect to the ES energy of the simulation results obtained with CST EM STUDIO®2012 and the BCDG method either with the *cell merging method* or the *adaptive approximation order method* and the additional *DG hybridization*. The square symbols in Figure 4.7 represent the relative errors and the lines are obtained by a least square fit through the relative errors for the applied approximation orders $p = 1, 2, 3$. In contrast to the convergence plots presented in Figure 4.3-4.6, the relative error is plotted with respect to the number of DOF. Plotting the relative error with respect to the grid parameter h is not feasible in this case, since the simulations in CST EM STUDIO®2012 are based on tetrahedral meshes with curved elements, whereas the BCDG method is based on structured hexahedral meshes. The relative errors in Figure 4.7 are illustrated using a log-log scale. It enables us to represent the rate of convergence with respect to the DOF as straight lines. The gradient or slope of the lines R represents the rate of convergence. Therefore, the convergence with respect to the relative error is approximately of order $\mathcal{O}(DOF^{-R})$.

The coloured solid lines represent the line fit through the relative errors of the simulation results obtained with CST EM STUDIO®2012 for $p = 1, 2, 3$. Note that the applied curved element order p_{ce} corresponds to the applied approximation order, i.e. $p = p_{ce}$. The dashed lines in Figure 4.7 are fitted lines through the relative errors of the BCDG method for the *adaptive approximation order method* and the *cell merging method* for $p = 1, 2, 3$.

As Figure 4.7 shows, the relative errors of the BCDG method for approximation order $p = 2$ and $p = 3$ are lower than the errors of the CST EM STUDIO®2012 simulations. For $p = 1$ the relative errors of CST EM STUDIO®2012 are lower than the relative errors of the BCDG method. More importantly, the rates of convergence of the BCDG method are higher than the rates of convergence of the simulations with CST EM STUDIO®2012 for $p = 1, 2, 3$. This result demonstrates that the BCDG method is more accurate and, therefore, more effective with respect to the number of DOF.

Next, we would like to present the results of a comparison study with respect to the computational costs. To make the results comparable across methods, we performed all simulations on the same computer. The computational costs are measured in seconds s. Note that the computational time is split into two times, the assembly time and the solver time. The solver time is plotted in Figure 4.8, whereas the assembly time is shown in Figure 4.9. In both figures, the square symbols represent the measured time and the lines are fitted through the measured data using a log-log scale. The slope rates R are listed in the legend of the figures. Figure 4.8 and 4.9 show that the solver times as well as the assembly times of the BCDG method are higher than the corresponding solver and assembly times of the CST EM STUDIO®2012 simulations, respectively. Furthermore, the slope rates of the BCDG method are higher than the slope rate of the CST EM STUDIO®2012 simulations results. This implies that the simulation with CST EM STUDIO®2012 is more efficient in terms of computational costs.

Since we mainly focussed on the accuracy of the BCDG method, this result is not surprising. The computational costs were of minor importance in the development process of BCDG method. In other words, the BCDG method is not optimally implemented with respect to computational time. However, the BCDG method could be improved in various dimensions to lower the computational costs. In particular, the computational costs of the quadrature can be reduced substantially. In the current implementation we use a quadrature from the open source geometry kernel OCCT which is computationally costly. In future work, more emphasis should be put on

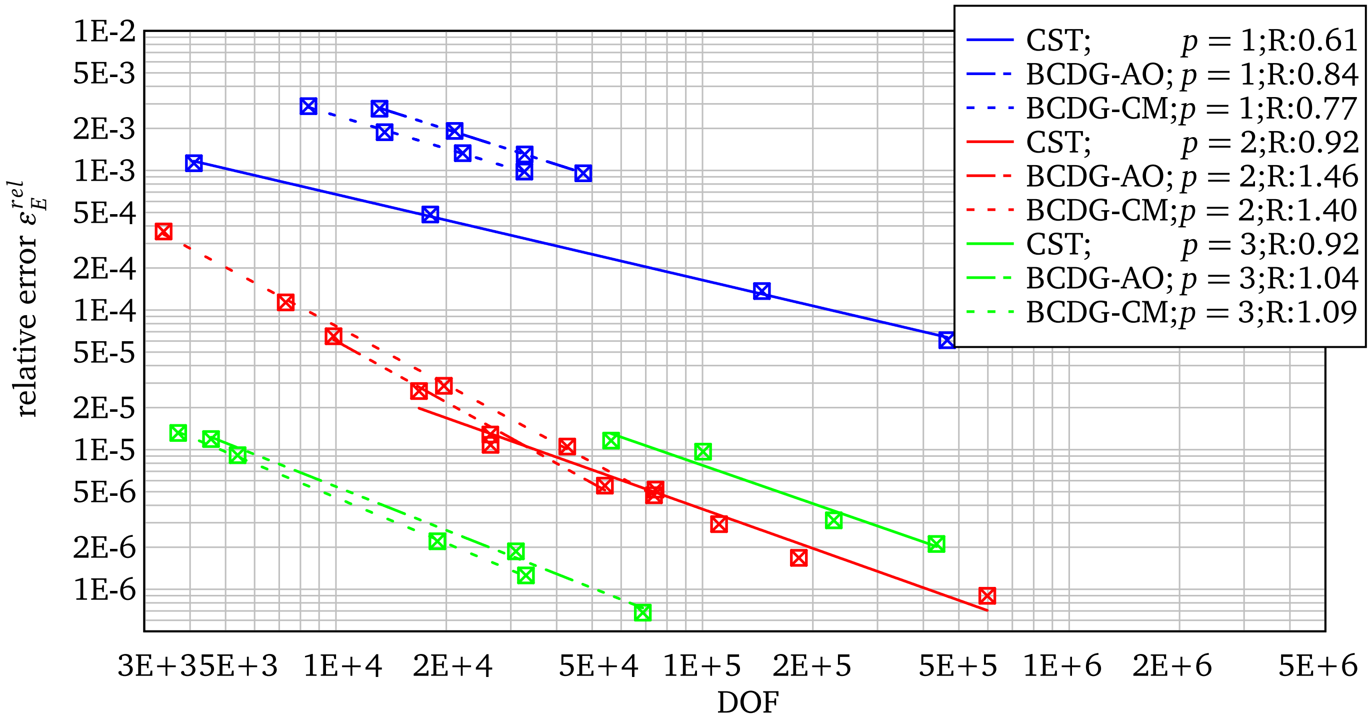

Figure 4.7.: ES energy comparison study - Relative errors obtained from CST EM STUDIO®2012 and the BCDG method combined with either the *cell merging method* or the *adaptive approximation order method* and the additional *DG hybridization*.

the development of a fast quadrature rule, especially if the method is coupled with a CFD solver.

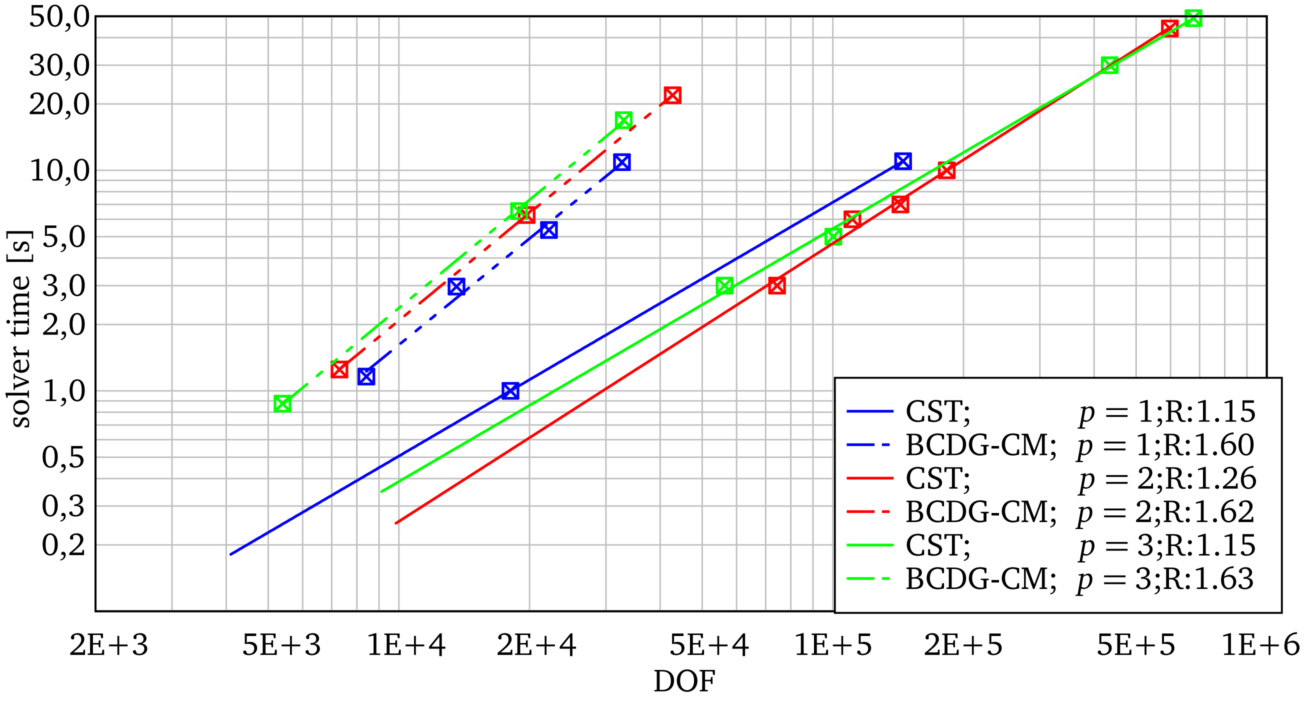

Figure 4.8.: Solver time results obtained with CST EM STUDIO®2012 and the BCDG method combined with the *adaptive approximation order method*.

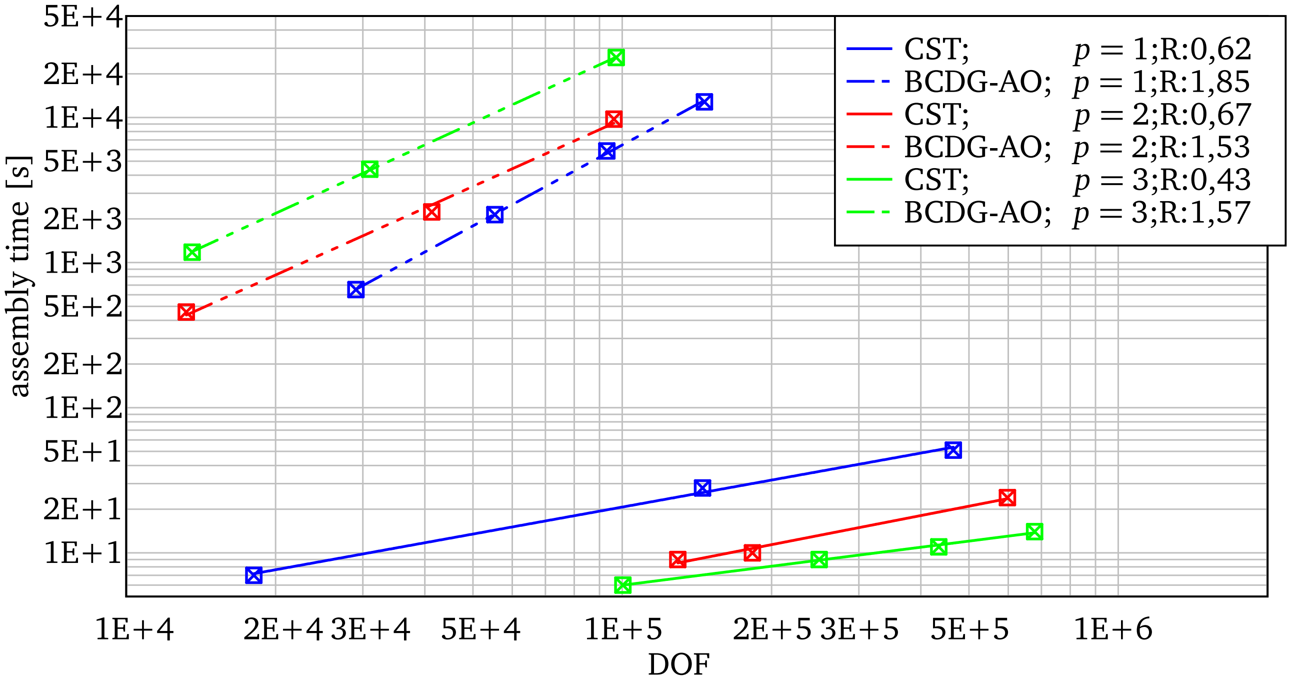

Figure 4.9.: Assembly time results obtained with CST EM STUDIO®2012 and the BCDG method combined with the *adaptive approximation order method*.

To conclude, the comparison studies of the simulation results obtained with CST EM STUDIO®2012 and the BCDG method show that the BCDG method performs slightly better with respect to the DOF than the ES solver of CST EM STUDIO®2012. The overall relative errors are lower for the approximation order $p = 2, 3$ and the convergence rates are higher for $p = 1, 2, 3$. Nonetheless, the BCDG method performs less well with respect to the computational costs, which implies that the BCDG method is not as efficient in terms of computational time as the ES solver of CST EM STUDIO®2012.

4.1.5 Impact of the adaptive approximation order and the cell merging method on the condition number

In Section 3.3.4, we introduced two supplementary methods for the BCDG method, the *adaptive approximation order method* and the *cell merging method*. These methods were developed to reduce the condition number of the system of linear equations of the DG formulation. In this section, we check whether these methods are indeed capable of reducing the condition numbers of the system of equations. Since the condition number indicates the sensitivity of the system to small disturbances [77] it is important to keep the condition number of our system of equations low.

In the following we present the condition numbers of the BCDG method with and without the additional *adaptive approximation order method* and the *cell merging method*. We thereby focus on the condition numbers for the ES cylindrical capacitor example. We use the same computational domain, material parameters and threshold values as presented in Section 4.1.1 and 4.1.3. The condition numbers are evaluated for the system matrix A of the system of linear equations in (3.114) which we presented in Section 3.2.5. The condition number of the matrix **A** is defined by (see, e.g. [77])

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\| \tag{4.10}$$

with

$$\|\mathbf{A}\| = \sup_{\mathbf{x}\in\mathbb{R}^N} \frac{\mathbf{Ax}}{\mathbf{x}}, \tag{4.11}$$

where **A** is given by

$$\mathbf{A} = \left(-\mathbf{G}^T \mathbf{I}_\epsilon \mathbf{M}^{-1}\mathbf{G} + \mathbf{S}\right). \tag{4.12}$$

The norm $\|\cdot\|$ denotes the standard Euclidean norm.

Figure 4.10 illustrates the condition numbers of the system matrix A for different grid parameters h and approximation order $p = 2$. The condition number of A is calculated before any preconditioner is applied to the matrix. The red square symbols mark the condition numbers of A without the *adaptive approximation order method* and *cell merging method*. The blue and green square symbols illustrate the condition number of the BCDG method combined with either the *adaptive approximation order method* or the *cell merging method*, respectively.

Figure 4.10 indicates that the condition numbers for the BCDG method without the additional methods are significantly higher than the values for the BCDG method with the additional methods. This implies that the condition number is reduced substantially if the BCDG method is used with either the *cell merging method* or the *adaptive approximation order method*. Furthermore, note that the condition numbers remain at the same order of magnitude with increasing grid parameter h.

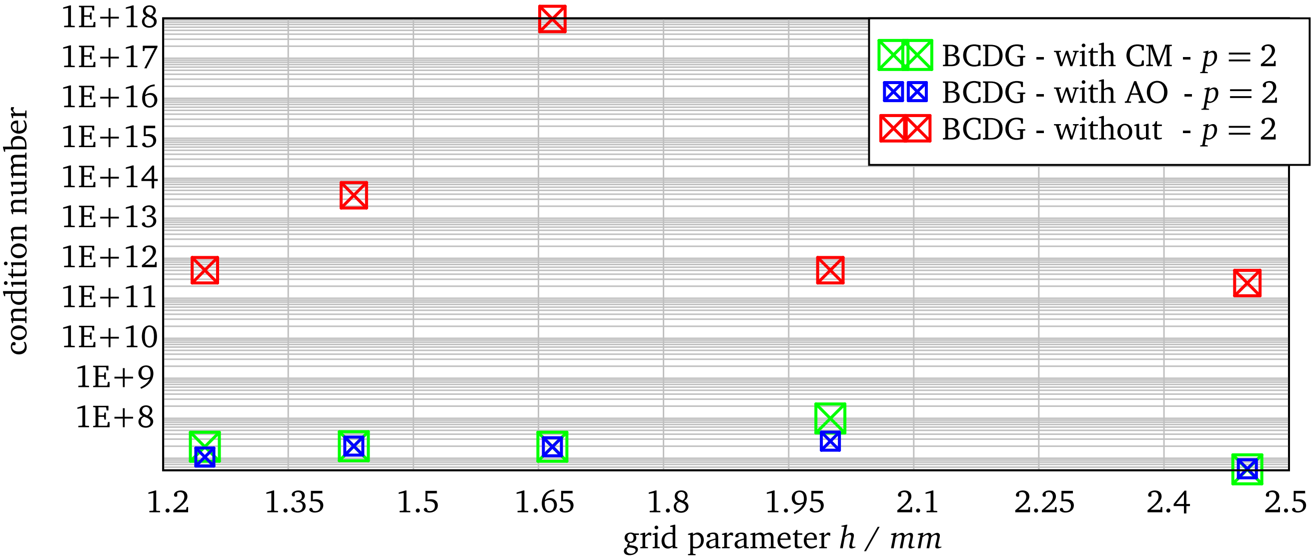

Figure 4.10.: Impact of the *adaptive approximation order method* and the *cell merging method* on the condition numbers.

To complete the discussion of the additional methods, we assess their impact on the convergence behaviour of the iterative solver. As it turns out, either the *adaptive approximation order method* or the *cell merging method* are necessary to obtain convergence of the iterative solver for the BCDG method. If the BCDG method is used without the *adaptive approximation order method* or the *cell merging method*, the iterative solver doesn't converge to the solution of the system of linear equations. By contrast, it converges for all grid parameters if the BCDG method is applied with either the *adaptive approximation order method* or the *cell merging method*. This demonstrates that the BCDG method has to be applied with either the *adaptive approximation order method* or the *cell merging method* both to obtain accurate results and to avoid ill-conditioning.

4.1.6 Impact of the DG hybridization method on the number of DOF

In Section 3.3.5 we proposed a *DG hybridization* method which transforms DG DOF in normal elements to FE DOF to reduce the overall high number of DG DOF. The application of this *DG hybridization* can only be justified if the number of DOF can be reduced significantly without lowering the accuracy of the BCDG method. For this reason, we check whether these two objectives can be achieved in this section. We apply the BCDG method combined with the *cell merging method* with and without the *DG hybridization* to the ES cylindrial capacitor field problem and compare the resulting number of DOF in both cases. The material and computational domain parameters are set to the values presented in Section 4.1.1 and 4.1.3. Furthermore, we evaluate the relative error to examine the accuracy difference with respect to the number of DOF. The resulting numbers of DOF of the BCDG method with and without the *DG hybridization* are shown in Figure 4.11. The symbols represent the exact numbers of DOF with respect to the grid parameter h. The lines are fitted through the DOF numbers. Note that the blue lines represent the numbers of DOF for approximation order $p = 1$, whereas the red lines denote approximation order $p = 2$. The number of DOF of the dashed lines are obtained by applying the BCDG method without the *DG hybridization*, whereas the number of DOF of the solid line are obtained with the *DG hybridization*. As demonstrated in Figure 4.11, the reduced number of DOF grows with decreasing grid parameters. Since the geometry model doesn't change in the computational setup, a smaller grid parameter implies a higher proportion of normal elements. This, in turn, results in a higher number of reduced DOF.

As a next step, we show that the relative errors of the BCDG method with and without the *DG hybridization* are approximately of the same order of magnitude. Figure 4.12 illustrates the relative error of the ES cylindrical capacitor field problem. The relative error is calculated as proposed in (4.3) for different grid parameters h and the approximation orders $p = 1, 2$. The symbols represent the resulting relative errors of the BCDG method with and without the *DG hybridization*. The lines are fitted lines through the relative errors. The red and blue coloured symbols and lines denote the relative error of the BCDG method with and without the *DG hybridization* for $p = 1$, whereas the green and pink coloured symbols and lines illustrate the results for $p = 2$, respectively. Figure 4.11 illustrates that the difference between the relative errors with and without the *DG hybridization* is very small. The convergence rate with and without the *DG hybridization* is exactly the same. This result demonstrates that the application of the *DG hybridization* is justified since the number of DOF can be reduced substantially without influencing the accuracy of the method. Moreover, since the DOF reduction only involves a few matrix-matrix and matrix vector multiplications, the additional computational costs are relatively small.

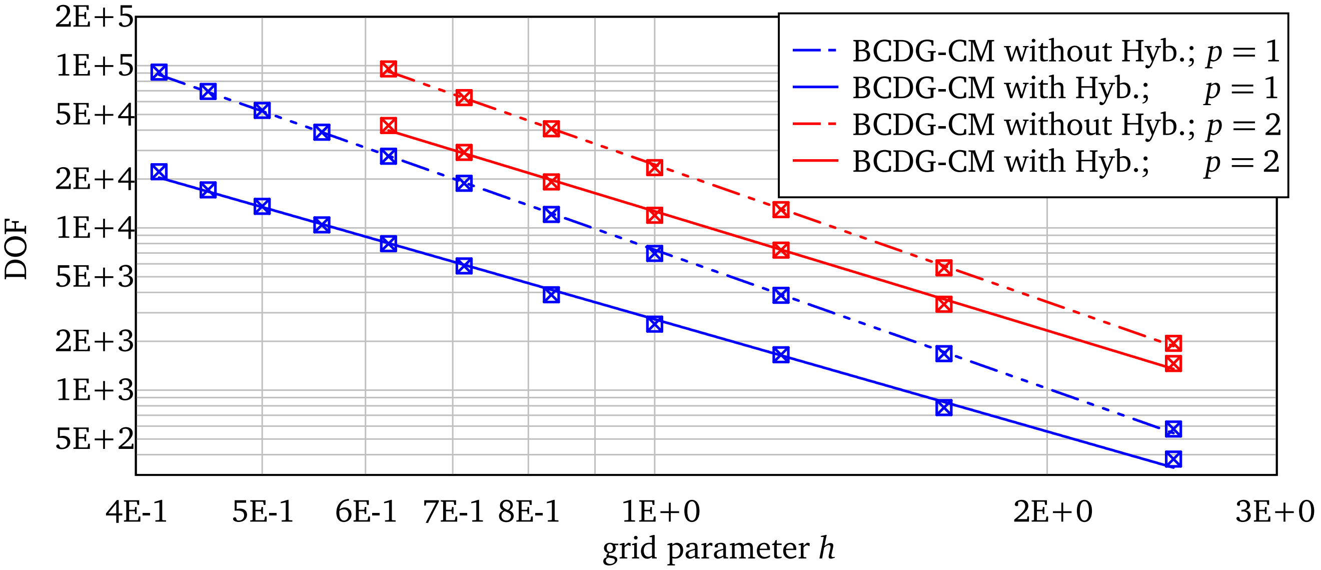

Figure 4.11.: Number of degrees of freedom of the BCDG method combined with the *cell merging method* with and without the *DG hybridization* method.

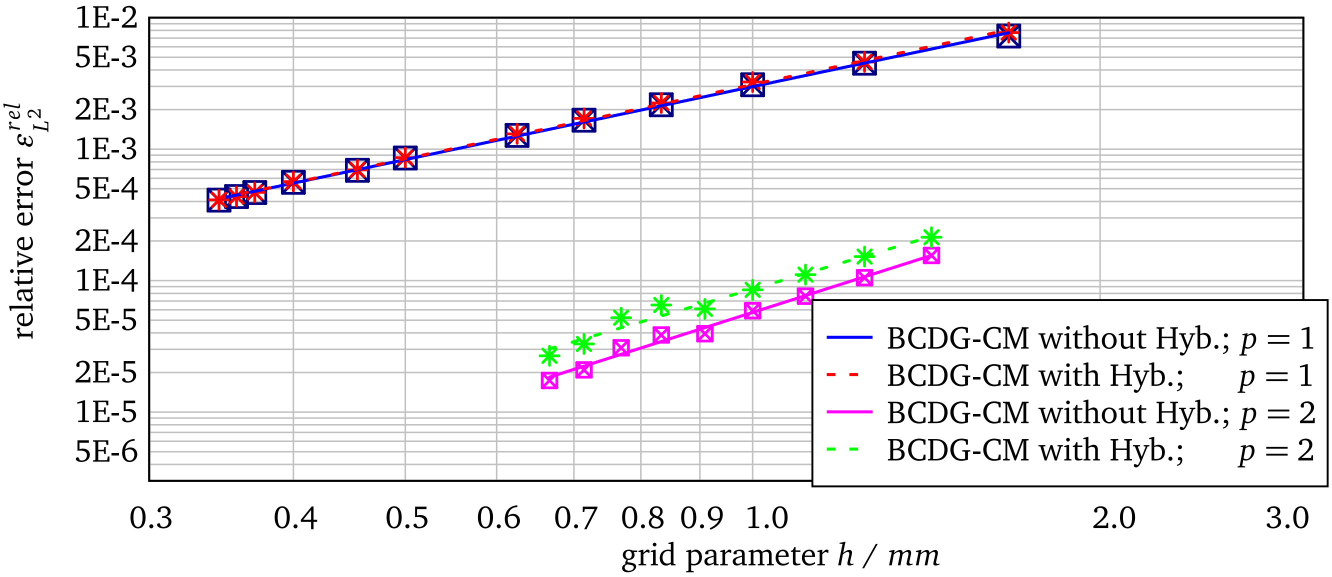

Figure 4.12.: Relative error measured in the L^2-norm of the BCDG method combined with the *cell merging method* with and without the *DG hybridization method.*

4.2 Simulation example of a water droplet on a high voltage insulator

In this section, we present the simulation results for the EBCDG method introduced in Section 3.6. As we have stated in Section 3.6 the EBCDG method is an extension of the BCDG method to field problems which comprise materials that move over time. Therefore, we apply the EBCDG method to an example of a water droplet oscillating on the insulation layer of a high voltage insulator. As shown in Section 2.2.2.2, this example represents an EQS field problem.

This section is organized as follows: First, the modelling details of the water droplet example are introduced. Then, the simulation results of the EBCDG method for the water droplet example are shown for several grid parameters h at various points in time. Finally, a convergence study of the EBCDG method is presented. Since there is no analytical solution known for the water droplet example, we cannot show convergence with respect to an analytical solution. However, we are able to conduct a convergence study with respect to the energy and power dissipation of the EBCDG method for the application example.

4.2.1 Modelling of the EQS field problem of an oscillating water droplet lying on the insulation layer of a high voltage insulator

The example of a water droplet oscillating on a planar insulation layer is shown in Figure 4.13. The blue material domain Ω_W represents the oscillating water droplet and the green material domain Ω_I a part of the plane insulation layer. The grey coloured sheets represent the parallel plate electrodes which are placed below and above the water droplet and the insulation layer. The third material considered in the simulation is the air between the insulation layer, the water droplet and the upper plate electrode. This domain is denoted as Ω_A.
A sinusoidal voltage excitation with an amplitude of $1kV$ and an operating frequency of 50HZ is applied to the electrodes. This voltage excitation is used to create an exemplary vertical potential and electric field distribution which might occur on the insulation layers of high voltage insulators.
Since the EBCDG method is not yet coupled with a fluid dynamic solver, we are not able to evaluate the motion of the water droplet during the simulation. Therefore, we use a given water droplet motion and apply our EBCDG method to this given motion of the water droplet. The use of a given artificial water droplet motion has the advantage that the results of the EBCDG method can be presented without accounting for additional external influences of CFD solvers or coupling effects on the simulation result.

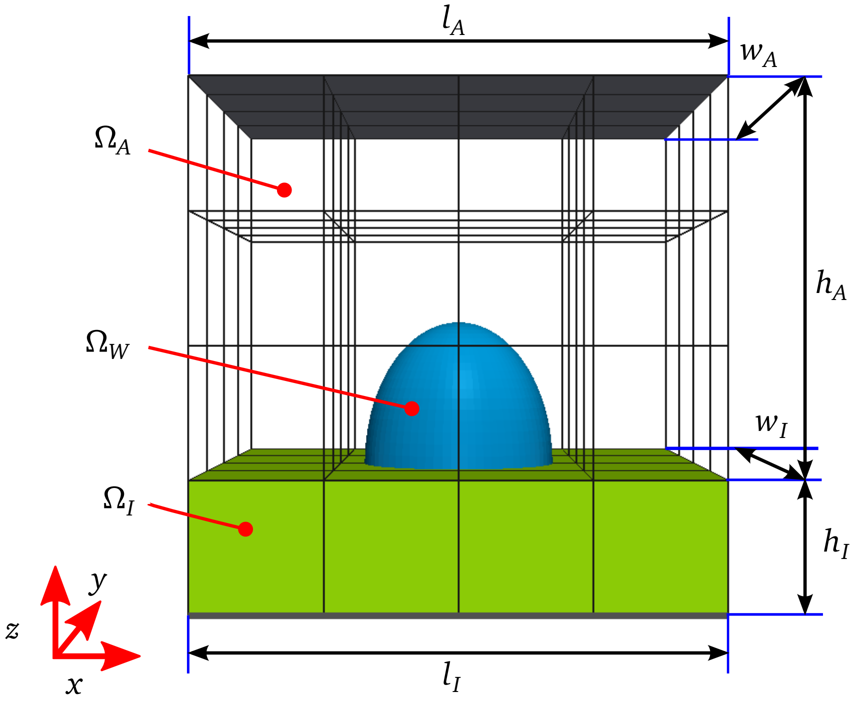

Figure 4.13.: Cartesian grid containing an arbitrarily shaped water droplet on an insulation layer surrounded by air. The grey coloured sheets represent the parallel plate electrodes.

The computational domain of the application example is given as $\Omega = \Omega_I \cup \Omega_A \cup \Omega_W$ and has the size length $l = l_x = l_y = l_z = 10.0mm$ in each axis direction. The material domains, Ω_I, Ω_A and Ω_W are all modelled as B-Rep shapes using OCCT [2]. The insulation layer Ω_I is represented by a hexahedral B-Rep shape having the side length $l_I = 10.0mm$, width $w_I = 10.0mm$ and height $h_I = 2.5mm$. The oscillation of the water droplet Ω_W is modelled by different B-Rep shapes at pre-specified points in time. These points in time are defined by $t^n = t_0 + n\Delta t$, where $n = 0, 1, \dots, N$ and Δt denotes the time interval $\Delta t = (t_N - t_0)/N$. The water droplet shapes at the pre-specified points are ellipsoids which are divided by the plane of symmetry in z-direction. Figure 4.14 shows two exemplary water droplet shapes at two different points in time.

The water droplet ellipsoids are described using the standard equation of ellipsoids,

$$\frac{(x-5)^2}{a^2} + \frac{(y-5)^2}{b^2} + \frac{(z-2.5)^2}{c^2} = 1, \tag{4.13}$$

where a, b and c are the semi-principal axes[14]. The semi-principal axes at each pre-specified points in time t^n are defined as follows:

$$\begin{aligned} a &= 2.0mm, && (4.14)\\ b &= 2.0mm, && (4.15)\\ c(t^n) &= (2.25 - 0.0125\sin(2\pi f\, t^n)k)mm, && (4.16) \end{aligned}$$

where $f = 50Hz$. Furthermore, the water droplet oscillation is represented by $N = 80$ water droplet shapes. These shapes replicate a possible vertical motion of water droplet on high voltage insulators.

The computational domain of the air Ω_A is characterized by length $l_A = 10.0mm$, width $w_A = 10.0mm$ and height $h_a = 7.5mm$. Since the domain of the air surrounds the water droplet, it is obtained by applying the *Cut Boolean operation* provided by OCCT [2] to a hexahedral B-rep shape with the physical dimensions l_A, w_A and h_A and the water droplet B-rep shapes at the pre-specified points in time. Therefore, the computational domain of the air is also modelled by different B-rep shapes at the pre-specified points in time.

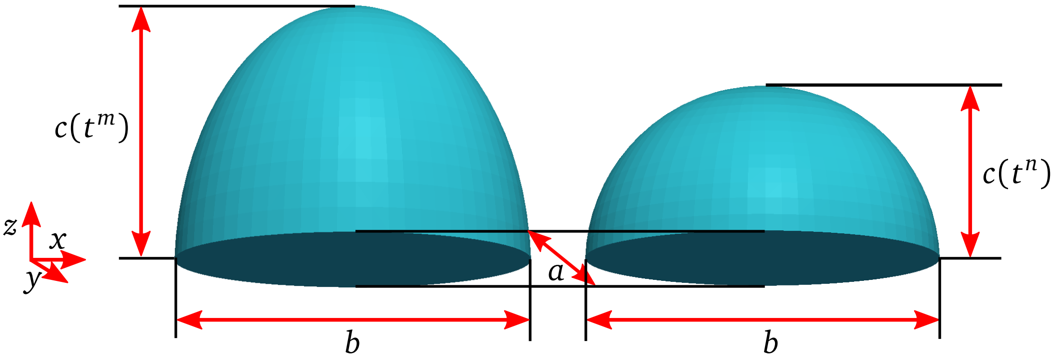

Figure 4.14.: Two different water droplet shapes.

Next, we introduce the material parameters of the computational domains. The permittivity and conductivity parameters of the water droplet are set to $\epsilon_W = 80.0F/m$ and $\kappa_W = 5.5$E-8 S/m, respectively. The material parameters of the insulation layer and the air are set to $\epsilon_I = 2.8F/m, \kappa_I = 3.3$E-15 S/m and $\epsilon_A = 1.0F/m, \kappa_A = 3.3$E-15 S/m, respectively.

The boundary $\Gamma = \partial\Omega$ of the computational domain Ω is partitioned into boundaries with Dirichlet and Neumann boundary conditions. Dirichlet boundary conditions are applied to the boundary faces which are normal to the z-direction. These faces represent the parallel plate electrodes. The voltage excitation at the electrodes is enforced by the Dirichlet boundary conditions. More precisely, the potential value $\phi_{D1}(t) = 1000\sin(2\Pi f t)V$ with $f = 50Hz$ is applied to the upper face, whereas $\phi_{D2}(t) = 0V$ is applied to the lower face. Furthermore, we assume that the length L of parallel plate electrodes is long compared to the distance between the electrodes l_z. This implies that the electric field in x- and y-direction vanishes on the boundary faces which are normal to the x- and y-direction. Therefore, the vanishing electric field in normal direction on these boundary faces are modelled by homogeneous Neumann boundary conditions.

Furthermore, we apply the EBCDG method with the *adaptive approximation order method*. The threshold values of the *adaptive approximation order method* are chosen as presented in Section 4.1.3.3. We decided to implement the *adaptive approximation order method* instead of the *cell merging method*, because it yields similar results with respect to accuracy and is easier to implement.

The initial conditions of the EBCDG method are set to $\phi(t_0 = 0) = 0$ and the time interval of the simulation is given as $[t_0 = 0s, t_e = 0.1s]$. This time interval is equivalent to five periods. Since the initial conditions are set to zero, they induce an additional potential solution term which is approximately of order e^{-t/τ_e}. At $t = 0.08s$ the influence of the initial conditions is small enough and can be neglected. For this reason, we only consider simulation results obtained in the time interval $[t_a = 0.08s, t_e = 0.1s]$.

The Θ-method with $\Theta = 1$ (Backward Euler method) is applied to the resulting semi-discrete weak DG formulation in (3.177)-(3.177). As mentioned before, we decided to use the Backward Euler method since it simplifies the EBCDG method introduced in Section 3.6. There are two further reasons for applying the Backward Euler method to the semi-discrete weak DG formulation. First, the semi-discrete weak DG formulation of the EBCDG method on the presented application example is considered as a stiff ODE. This is because the ratio of the maximum and minimum eigenvalues of our system of ODEs is given by 5.83E+05. If the water droplet conductivity is lowered, the ratio of the maximum and minimum eigenvalue of our system is even higher. If we assumed a conductivity value of rain water droplets ($\kappa_W = 5.5\text{E-}1\ S/m$) the ratio would be of order 5.83E+13. Since the Backward Euler method is an implicit method, it performs in general better on stiff ODE than the explicit Euler method ($\Theta = 0$). Furthermore, all eigenvalues of our system are negative. This implies that the implicit and L-stable Backward Euler method ($\Theta = 1$) is better suited than the Crank-Nicolson method ($\Theta = 1/2$). Finally, the resulting system of linear equations is solved using the direct solver SuperLU_Dist provided by PETSc [3] (see Section 3.5 for more details).

4.2.2 EQS simulation results of the EBCDG method for the water droplet example

In this section, we present the simulation results for the potential and electric field solutions of the water droplet example for the water droplet oscillation defined in Section 4.2.1. We show the potential and electric field distribution of the EBCDG method for 512 elements combined with approximation order $p = 3$ and for 4096 elements combined with approximation order $p = 2$ at two points in time, $t = 0.085s$ and $t = 0.095s$. At these points in time the sinusoidal excitation and the oscillation reach the maximum and minimum amplitude values, respectively.

Figure 4.15 shows the potential simulation results of the EBCDG method at time $t = 0.085s$ and $t = 0.095s$ on a cut-plane at $x = 5mm$. Note that the cut-plane shows the potential distribution at the water droplet's x-axis of symmetry.

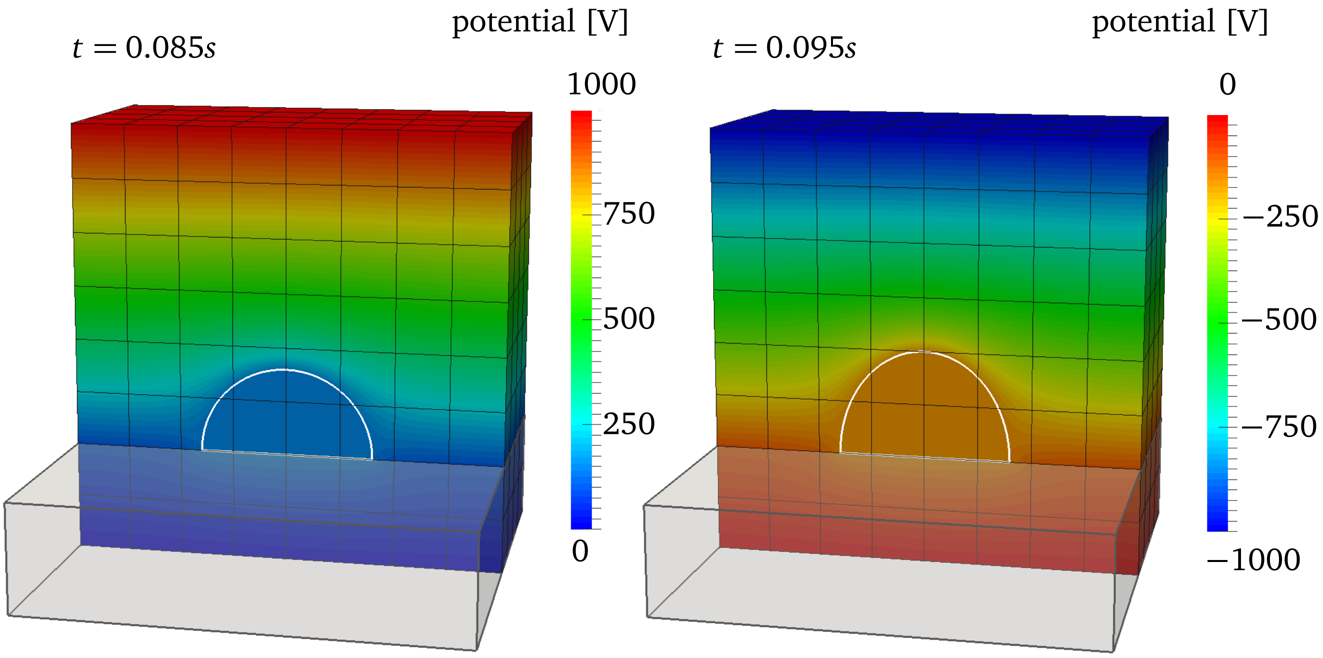

Figure 4.15.: Potential distribution of the EBCDG method for 512 elements and approximation order $p = 3$ at time $t = 0.085s$ and $t = 0.095s$.

Furthermore, the electric field distribution on the cut-plane at $x = 5mm$ are presented in Figure 4.16. Both figures demonstrate that we obtain potential and electric field distributions with

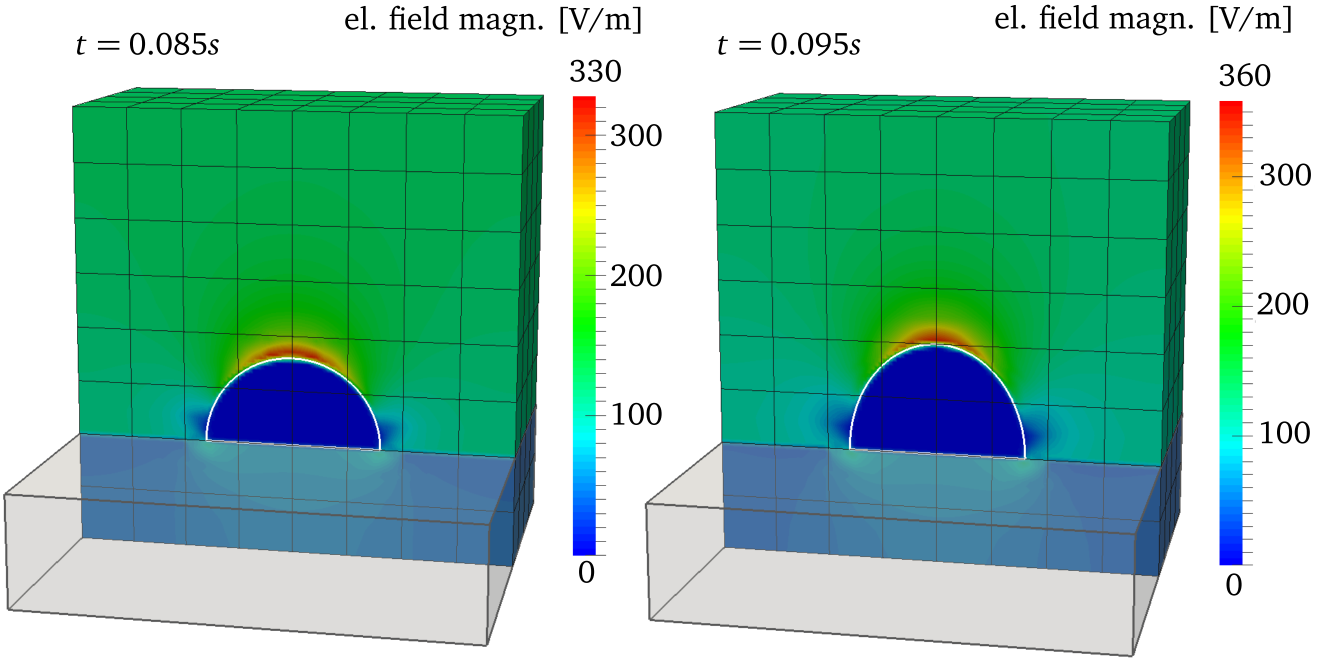

Figure 4.16.: Electric field distribution of the EBCDG method for 512 elements and approximation order $p = 3$ at time $t = 0.085s$ and $t = 0.095s$.

a high resolution even on a coarse mesh with 512 hexahedral elements. This is due to the application of the cut-cell method and the high order accuracy of EBCDG method.

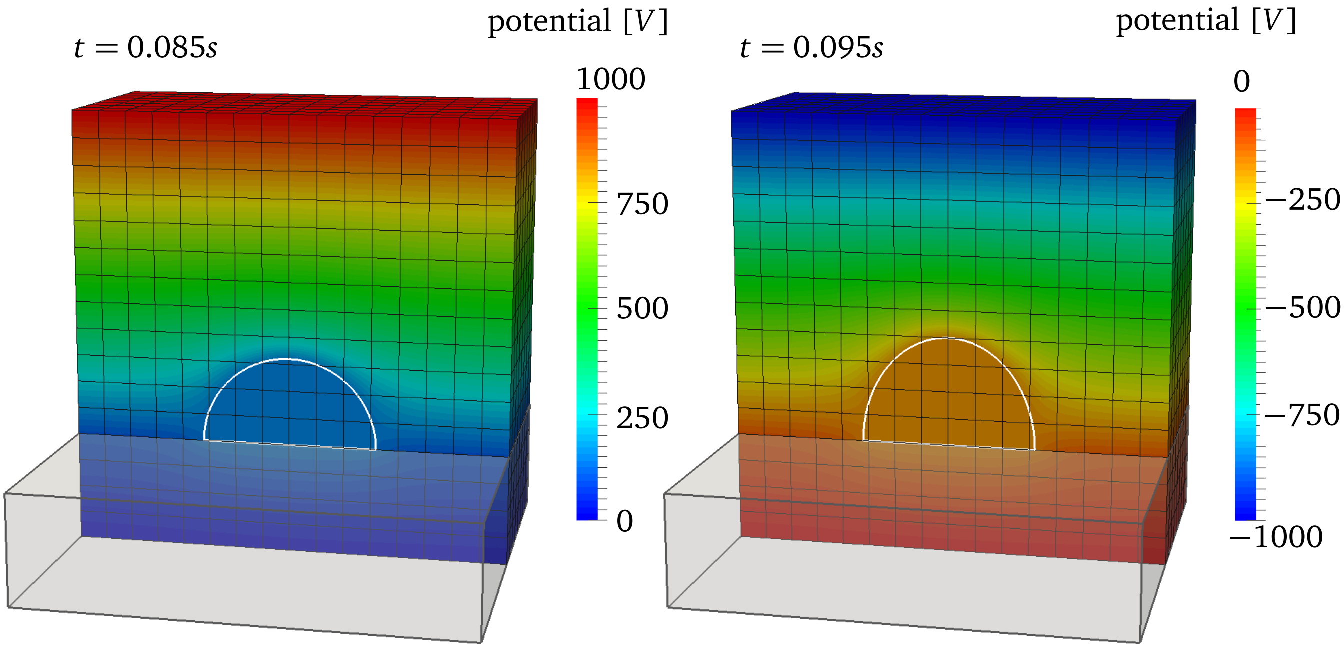

Figure 4.17.: Potential distribution of the EBCDG method for 4096 elements and approximation order $p = 2$ at time $t = 0.085s$ and $t = 0.095s$.

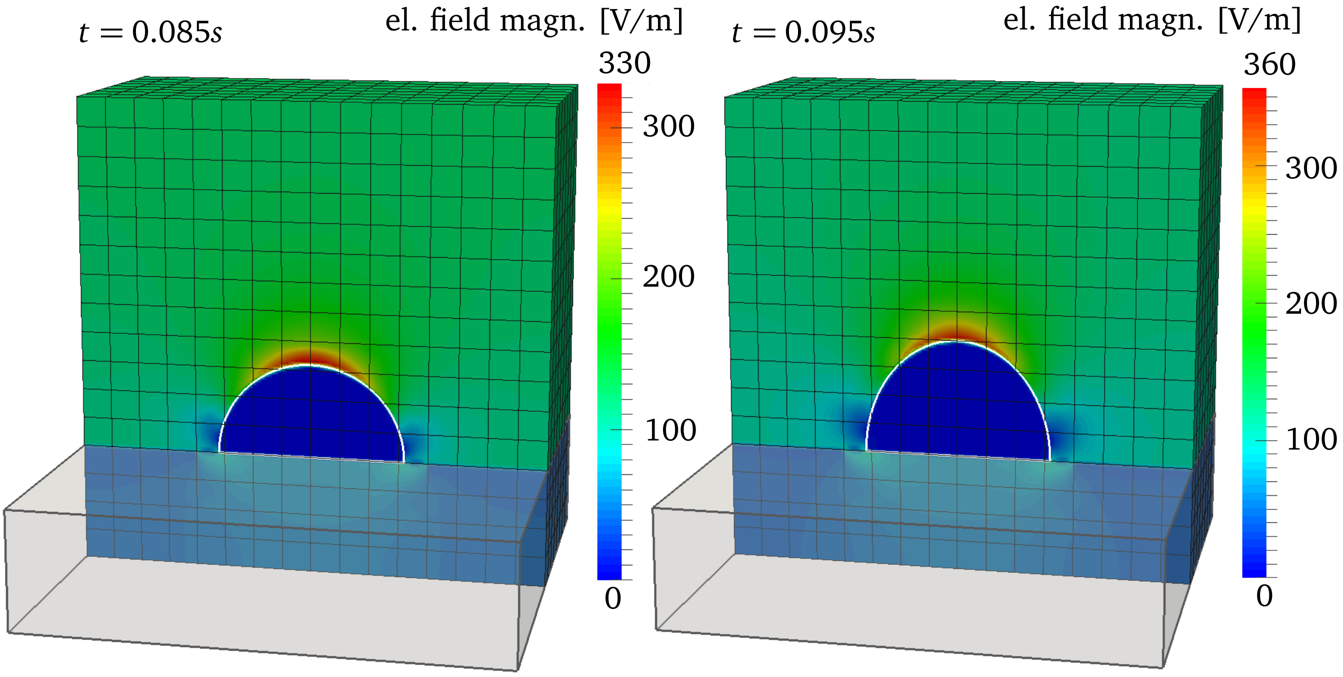

Figure 4.18.: Electric field distribution of the EBCDG method for 4096 elements and approximation order $p = 2$ at time $t = 0.085s$ and $t = 0.095s$.

Figure 4.17 and 4.18 present the potential and electric field distribution of the EBCDG method for 4096 elements and approximation order $p = 2$. As before, the potential and electric field distributions at $t = 0.085s$ and $t = 0.095s$ are shown on a cut-plane at $x = 5mm$.

Both figures illustrate that also in this case a high resolution potential and electric field distribution is obtained using the EBCDG method.

Next, we demonstrate that the electric field distribution of the simulations for 512 elements and $p = 3$ as well as for 4096 elements and $p = 2$ are of the same order of magnitude at $t = 0.085s$ and $t = 0.095s$. To illustrate this, we present two line plots of the electric field through the z-axis of the water droplet at $x = 5mm$ and $y = 5mm$.

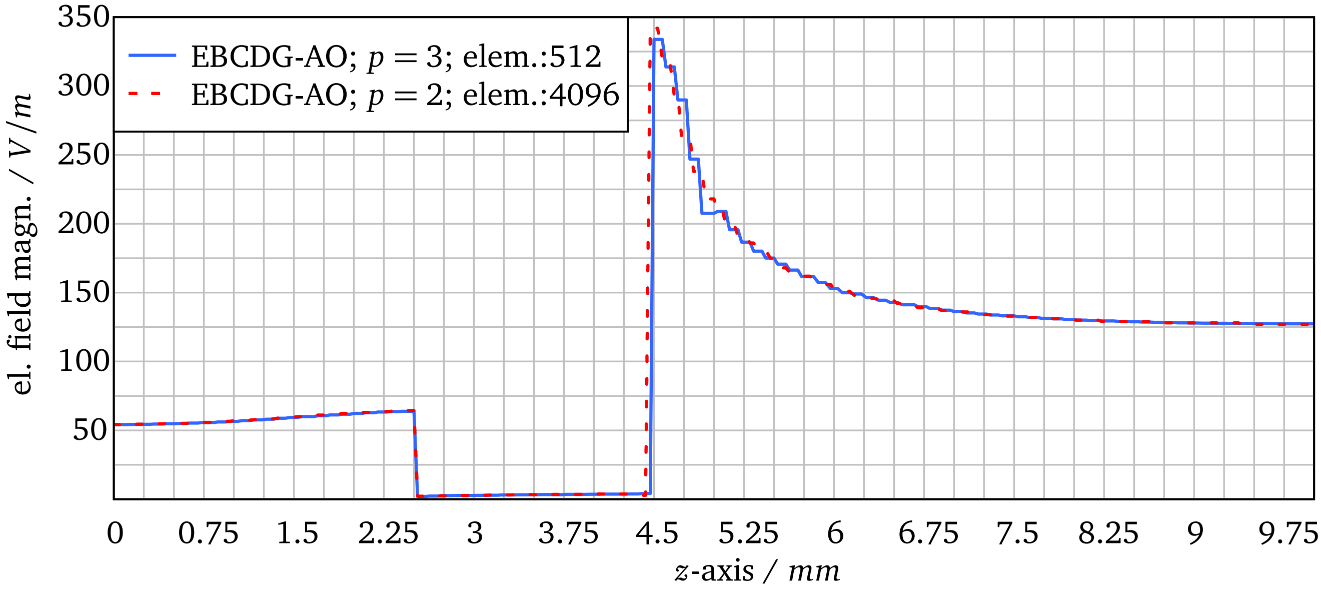

Figure 4.19.: Electric field plot over a vertical line through the droplet center at $x = 5mm$ and $y = 5mm$ in z-direction at time $t = 0.085s$.

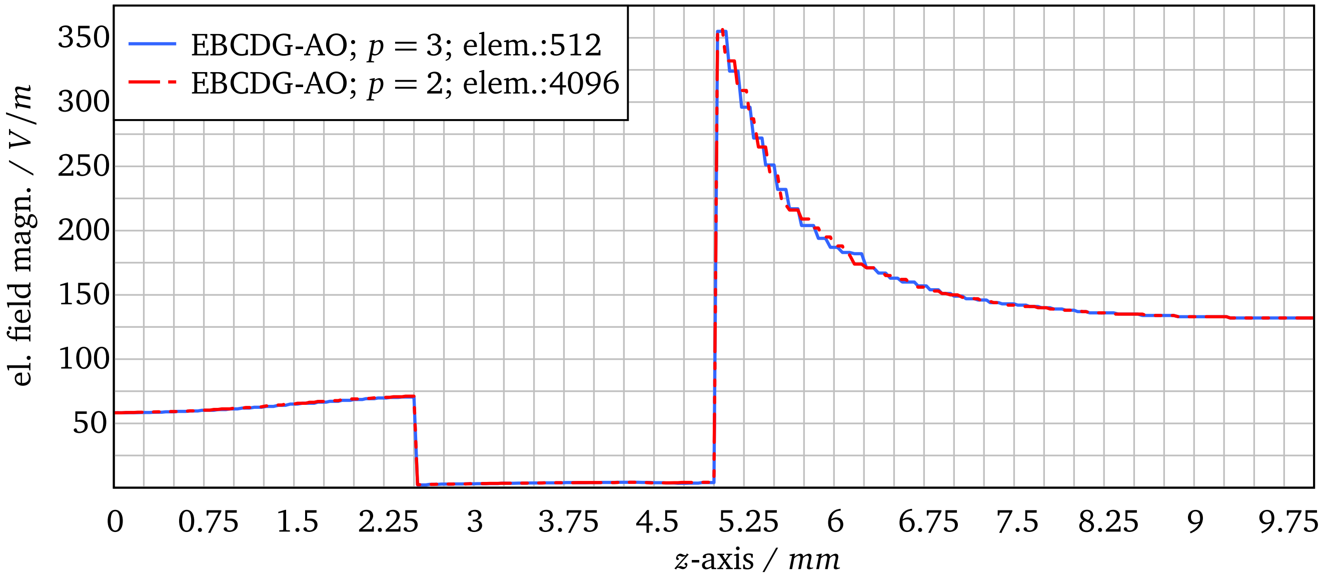

Figure 4.20.: Electric field plot over a vertical line through the droplet center at $x = 5mm$ and $y = 5mm$ in z-direction at time $t = 0.095s$

Figure 4.19 shows the electic field simulation results of the EBCDG method over a line at $x = 5mm$ and $y = 5mm$. The blue coloured solid line represents the electric field for 512 elements and $p = 3$, whereas the red coloured dashed line represents the electric field for 4096 elements and $p = 2$ at time $t = 0.085s$.

In Figure 4.20 the electric field simulation results of the EBCDG method at time $t = 0.095s$ over the same line are shown. As before, the blue coloured solid line presents the electric field for 512 elements and $p = 3$, whereas the red coloured dashed line presents the electric field for 4096 elements and $p = 2$ at time $t = 0.095s$.

Both figures demonstrate that the electric fields are comparable across the two simulations with different meshes and approximation orders. These results represent an important first step towards the verification of the EBCDG method. To further validate the method, we show in the next section that the simulations converge to a specific energy level for increasing approximation orders p and decreasing grid parameters h.

4.2.3 Energy and dissipation convergence study

This section presents a convergence study of the EBCDG method for the water droplet example. Since no analytical solution for the water droplet example is known, we are not able to present the convergence study with respect to the analytical solution. However, we are able to show convergence with respect to the energy and power dissipation of the field problem for decreasing grid parameter h and increasing approximation order p. This is possible since we use a given artificial water droplet oscillation, which implies that the oscillation is the same for any grid parameter and any approximation order. Hence, the energy and power dissipation results of the EBCDG method on the water droplet example should converge to a particular energy and dissipation level for increasing grid parameters h and approximation orders p. Obviously, the convergence study with respect to the energy is not as convincing as a convergence study with respect to the analytic solution. Nevertheless, it verifies to some extent that the EBCDG method solves the field problem correctly and accurately.

The energy and power dissipation of the simulation results are evaluated as follows:

$$W_e = \frac{1}{2}\int_{t_a}^{t_e}\int_{\Omega_h} w_e dV dt = \frac{1}{2}\int_{t_a}^{t_e}\int_{\Omega_h} \mathbf{E}_h \cdot \mathbf{D}_h dV dt, \tag{4.17}$$

$$P_d = \frac{1}{2}\int_{t_a}^{t_e}\int_{\Omega_h} P_d dV dt = \frac{1}{2}\int_{t_a}^{t_e}\int_{\Omega_h} \mathbf{E}_h \cdot \mathbf{J}_h dV dt, \tag{4.18}$$

where w_e and P_d denote the electric energy and power dissipation density which we introduced in Section 2.2.2.3. To better illustrate the convergence to a specific energy and dissipation level, we have decided to integrate the energy and power dissipation results over the time interval $[t_a = 0.08s, t_e = 0.1s]$.

Figure 4.21 shows the EQS energy simulation results which are obtained for different grid parameter h and approximation orders p. The blue symbols represent approximation order $p = 1$, the red symbols $p = 2$ and the green symbols $p = 3$. Figure 4.21 demonstrates that the energy simulation results converge to the energy level limit which is represented by the black dashed line for decreasing grid parameter h and increasing approximation orders p.

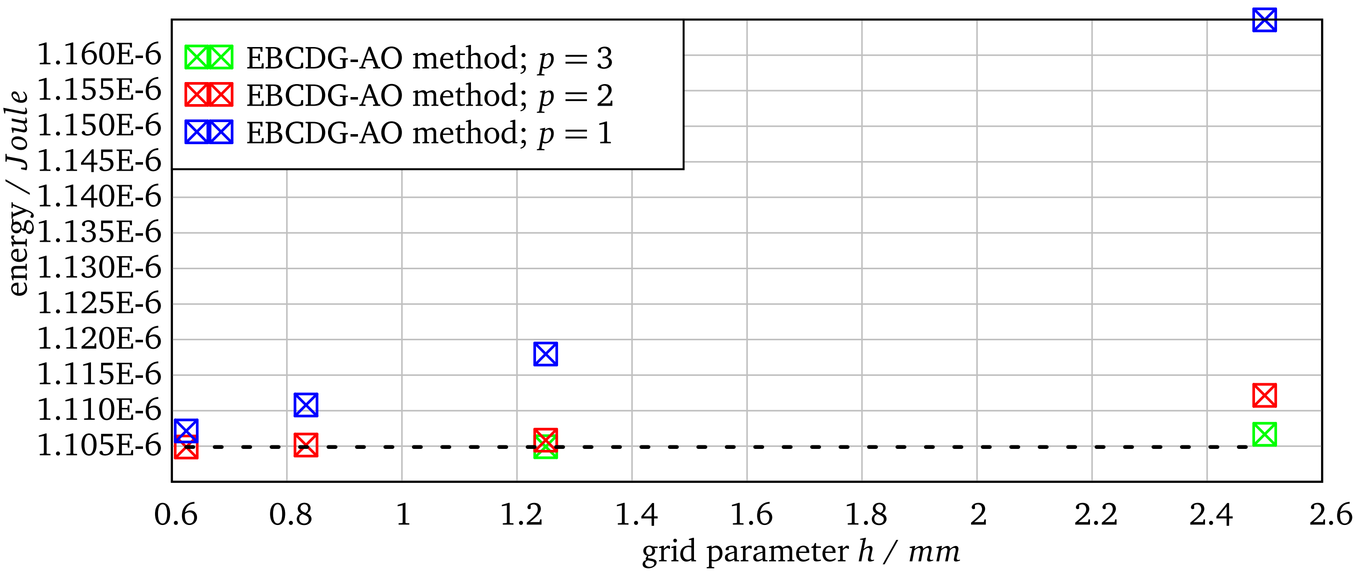

Figure 4.21.: Energy simulation results of EBCDG method with the *adaptive approximation order method* for the water droplet example.

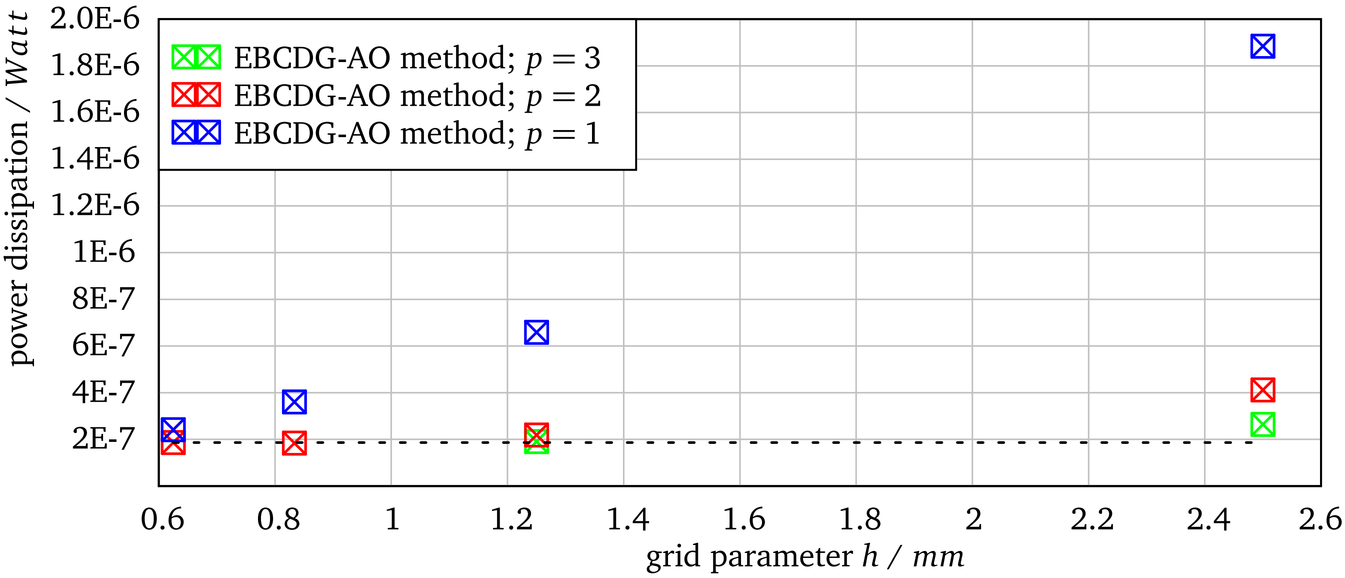

Figure 4.22.: Dissipation simulation results of EBCDG method with the *adaptive approximation order method* for the water droplet example.

The simulation results for the power dissipation are plotted in Figure 4.22. As before the coloured symbols indicate the different approximation orders and the black dashed line denotes the dissipation level limit. As shown in Figure 4.22 the simulation results for the power dissipation converge to the power dissipation limit for decreasing grid parameters h and increasing approximation orders p.

Next, we present the convergence study with respect to the time step Δt. Figure 4.23 shows the energy obtained for different time steps for grid parameter $h = 10/8$ and approximation order $p = 2$ and $p = 3$. The simulation results converge to a particular energy for decreasing time steps Δt. Note that the energy values for small time steps Δt correspond to the energy results shown in Figure 4.21 for grid parameter $h = 10/8$ and approximation order $p = 2$ and $p = 3$.

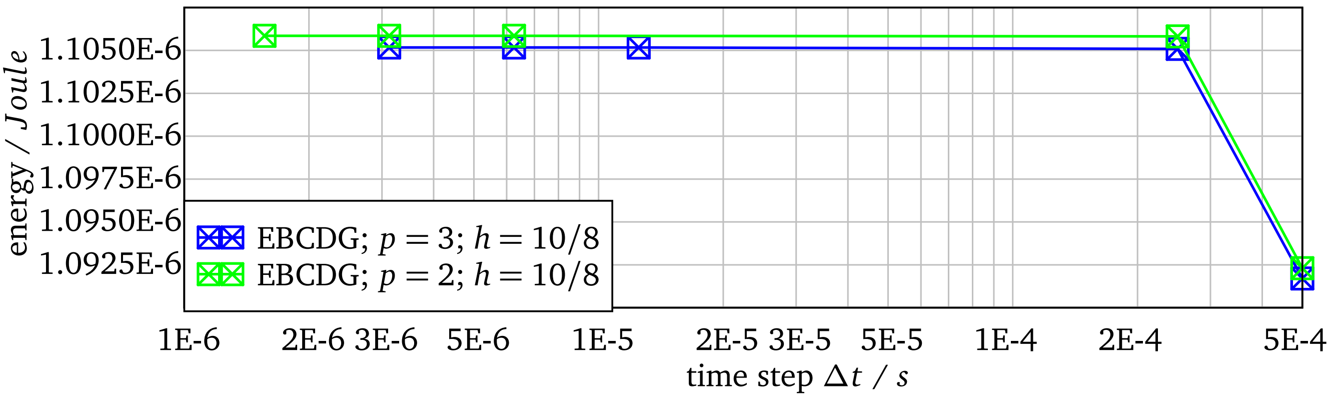

Figure 4.23.: Energy simulation results of the oscillating water droplet example with respect to the time step Δt.

To conclude, all presented convergence studies demonstrate that the energy and dissipation converge to a specific energy and dissipation level limit. These results are very promising and indicate that the EBCDG method can indeed be applied to field problems where materials move over time. As a next step, the EBCDG method could be coupled with a CFD solver to further examine the water droplet field problem.

However, a coupling with a CFD solver requires the application of meshes with more elements. So far, we are just able to present simulation results with the presented grid parameters $h \in [10/3, 10/16]$ and approximation orders $p = 1, 2, 3$. This has several reasons which are related to the implementation of the EBCDG method: First, the implementation needs a lot of memory. Second, the quadrature used in the EBCDG method is computationally costly. Third, our EBCDG method is not yet parallelized. However, all of the mentioned issues could easily be changed in future work.

Furthermore, we would like to mention that we use a conductivity value for the water droplet which is set to κ_W = 5.5E-8. We use this value since we wanted to show convergence with respect to the energy and power dissipation of the field problem. With a lower conductivity value it is not possible to show convergence to the energy and power dissipation level limit with the presented grid parameters and approximation orders. In this case, we have noticed that the difference in energy and power dissipation across the grid parameters h and approximation orders p becomes larger. This implies that the potential and electric field solution becomes more difficult to approximate. Therefore, meshes with more elements and higher approximation orders are needed for lower conductivity values.

Furthermore, note that the water droplet example becomes more stiff for lower conductivity values of the water droplet κ_W. As mentioned before, if we assume a conductivity value of rain water droplets (κ_W = 5.5E-1 S/m) the ratio between the minimum and the maximum eigenvalue of our system is of order 5.83E+13. However, with the conductivity value κ_W = 5.5E-8 S/m the ratio it is of order 5.83E+05. Therefore, we suggest to apply higher-order numerical methods for stiff ODEs to the semi-disrete weak DG formulation of the EBCDG method. Furthermore, the condition number of the resulting system of linear equations in (3.179)-(3.180) is very high. This implies that EBCDG method should be applied with higher threshold values for the *adaptive approximation order method* in future work.

Another research objective could be to embed the *cell merging method* into the EBCDG method. With the *cell merging method* it would be possible to use larger threshold values without having to assign lower approximation orders to too many geometrically small cut-cells.

5 Summary and Outlook

5.1 Summary

In this work we have presented the BCDG method and its extension, the EBCDG method. The BCDG method can be applied to ES and EQS field problems with non-moving materials, whereas the EBCDG method was developed for EQS field problems where materials move over time.

We first gave an overview on the theory of electromagnetics in Chapter 2. The Maxwell's equations and the ES and EQS approximations of the Maxwell's equations were discussed, which describe the physical behaviour of the field problems presented in this work. Furthermore, the conditions that justify an ES and EQS approximation of the Maxwell's equations were presented.

In Chapter 3, the fundamentals of the BCDG and the EBCDG method were introduced. Since the BCDG and the EBCDG method are based on the DG method, we presented the DG method in more detail and discussed its important properties as well as its advantages and disadvantages in Section 3.2. In Section 3.3, the general framework of the BCDG method was presented. The BCDG method comprises distinct approaches, namely the *cut-cell discretization approach*, the *adaptive approximation order method*, the *cell merging method* and the *DG hybridization*.

In Section 3.4, we presented the Θ-method, which is used to discretize the time derivatives in the semi-discrete weak formulation of the DG and the BCDG method. Furthermore, important features and properties of the Θ-method in the context of the DG and the BCDG method were discussed. In Section 3.6, the extension of the BCDG method to problems with moving materials was introduced. We described in greater detail how the EBCDG method is applied to moving material domains on a single fixed mesh.

In Section 4, various numerical simulation results of the BCDG and the EBCDG method were presented. In Section 4.1, we showed the numerical results of the BCDG method applied to the verification example of a cylindrical capacitor filled with two dielectric materials. The simulation results of the ES and EQS field problem of the cylindrical capacitor example show that optimal convergence rates are obtained for the BCDG method in combination with either the cell merging method or the adaptive approximation order method. However, for $p = 3$ the BCDG method in combination with the adaptive approximation order method yields slightly lower convergence rates. Furthermore, a comparison study of the BCDG method and the commercial software CST STUDIO SUITE®2012 [1] was presented in Section 4.1. The results of the BCDG method for the ES cylindrical capacitor example were benchmarked with results obtained from CST EM STUDIO®2012 [1]. This comparison study demonstrated that the BCDG method is slightly more accurate with respect to the number of DOF. However, the simulations with CST EM STUDIO®2012 are more efficient in terms of computational costs. Furthermore, the impact of the two additional methods - the cell merging and the adaptive approximation order method - on the condition number of the system matrices was assessed. As shown in Section 4.1, either of the two additional methods reduces substantially the condition number of the system for the ES cylindrical capacitor field example. At the end of Section 4.1 we illustrated that the DG hybridization significantly reduces the number of DOF without affecting the accuracy of the simulation results. In section 4.2, we presented the simulation results of the EBCDG method on the example of a water droplet oscillating on the insulation layer of a high voltage insulator. Since the EBCDG method is not yet coupled with a CFD solver, the water droplet motion had to be enforced artificially. However, the given artificial oscillation of the water droplet enabled us to present a convergence study with respect to the energy and power dissipation for the application example. The convergence study showed that the simulation results converge to a specific energy and power dissipation level both for increasing approximation orders p and decreasing grid parameters h. Furthermore, the potential and electric field distributions of the application example demonstrated that even on a coarse mesh a high resolution of the potential and electric field solution can be achieved.

5.2 Outlook

In Section 4.1, the comparison study illustrated that the simulations with CST EM STUDIO®2012 are more efficient in terms of computational costs. Especially the assembly time of the BCDG method is much higher than the assembly time obtained with CST STUDIO SUITE®2012. One reason for the lower performance with respect to computational time is the computationally costly quadrature used in this work. We believe that the assembly time can be reduced substantially if the present quadrature is modified or a more efficient quadrature is implemented. Since the BCDG method is implemented in a way that allows to easily change the geometry representation as well as the quadrature on the geometry representation, certain modifications with respect to the geometry and the quadrature could be performed easily and efficiently.

Another future research topic could be the coupling of the EBCDG method with a CFD solver for the Navier-Stokes equations. We are convinced that the boundary conformal approach could make further use of its inherent advantages if it is linked with a CFD solver using a similar cut-cell implementation. However, such a CFD solver is not yet available. Nevertheless, this coupling approach would enable the simulation of oscillating water droplets without having to use artificial oscillations of water droplets. This would further improve our understanding of the electromagnetic phenomenon.

A Shape functions for hexahedral elements

In this section, we present the detailed definition of the shape functions used in this work. These shape functions are proposed in [93] and span the local finite element spaces $\mathcal{S}(K_i)$ and $\mathcal{U}(K_i)$ (see Section 3.2.7). As mentioned in Section 3.2.7, the shape functions are defined on a hexahedral reference element given as $\hat{K} := \mathcal{H}[0,1]^3$. To obtain the shape functions for each element $K_i \in \Omega_h$, the reference element shape functions need to be mapped to each element. The affine map which is used to map the shape functions can be found in Section 3.2.7.

Note that the H^1-*conforming* and $H(div)$-*conforming shape functions* are defined using two trilinear functions λ_i and σ_i. These functions are given by (see, e.g. [93]):

$$\begin{array}{ll}
\lambda_1 = (1-x)(1-y)(1-z), & \sigma_1 = (1-x)+(1-y)+(1-z),\\
\lambda_2 = x(1-y)(1-z), & \sigma_2 = x+(1-y)+(1-z),\\
\lambda_3 = xy(1-z), & \sigma_3 = x+y+(1-z),\\
\lambda_4 = (1-x)y(1-z), & \sigma_4 = (1-x)+y+(1-z),\\
\lambda_5 = (1-x)(1-y)z, & \sigma_5 = (1-x)+(1-y)+z,\\
\lambda_6 = x(1-y)z, & \sigma_6 = x+(1-y)+z,\\
\lambda_7 = xyz, & \sigma_7 = x+y+z,\\
\lambda_8 = (1-x)yz, & \sigma_8 = (1-x)+y+z,
\end{array}$$

Furthermore, the H^1-*conforming* and $H(div)$-*conforming shape functions* are based on the Legendre and the Integrated Legendre polynomials [93]. The Legendre polynomials are defined in the three-term recurrence relation by

$$\begin{aligned}
l_0(x) &= 1, && \text{(A.1)}\\
l_1(x) &= x, && \text{(A.2)}\\
(n+1)l_{n+1} &= (2n+1)l_n(x)x - nl_{n-1}(x), n \geq 1 && \text{(A.3)}
\end{aligned}$$

or as $l_n(x) = \left(\frac{d}{dx}l_{n+1}(x) - \frac{d}{dx}l_{n-1}(x)\right)$ (see, e.g. [93]).
The Integrated Legendre polynomials $(L_n)_{2\leq i\leq p}$ are given as (see, e.g. [93])

$$L_n(x) := \int_{-1}^{x} l_{n-1}(\xi)d\xi \qquad \text{for } x \in [-1,1] \text{ and } n \geq 2. \tag{A.4}$$

Furthermore, note that the edge and face orientation is based on the global vertex indices. For edges E the edge orientation is specified by the vertex with the higher index number α which points to the vertex with the lower one. Therefore, edges are written as $E = [e_1, e_2]$ with

$$\begin{aligned}
e_1 &= \arg\max_{\alpha\in\{\alpha_1,\alpha_2\}} v_\alpha, && \text{(A.5)}\\
e_2 &= \arg\min_{\alpha\in\{\alpha_1,\alpha_2\}} v_\alpha, && \text{(A.6)}
\end{aligned}$$

where α_i states the index of vertex v_{α_i} [93].
The face orientation is given in a similar way. The faces are defined by $F = [f_1, f_2, f_3, f_4]$. The starting vertex f_1 is the vertex with the highest index number $f_1 := \arg\max_{\alpha \in \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}} v_\alpha$, vertex f_3 is the opposite of f_1 and f_2 is the vertex of the two remaining vertices with the higher index number, $v_{f_2} > v_{f_4}$.

Using the just introduced definition, the H^1-*conforming shape functions* are written as (see [93]):

Vertex-based functions:

$$\text{for } i = 1, ..., 8: \qquad \phi_i^V = \lambda_i \tag{A.7}$$

Edge-based functions:
For $m = 1, ..., 12$: For edge $E_m = [e_1, e_2]$, for $0 \le i \le p_{E_m} - 2$

$$\phi_i^{E_m} = L_{i+2}\left(\sigma_{e_1} - \sigma_{e_2}\right)\left(\lambda_{e_1} + \lambda_{e_2}\right), \tag{A.8}$$

Face-based functions:
For $m = 1, ..., 6$: For face $F_m = [f_1, f_2, f_3, f_4]$ and for $0 \le i, j \le p_{F_m} - 2$

$$\phi_{(i,j)}^{F_m} = L_{i+2}(\xi_F) L_{j+2}(\xi_F) \lambda_F, \tag{A.9}$$

with $\lambda := \sum_{\alpha=0}^{4} \lambda_{f_\alpha}$ and $(\sigma_{f_1} - \sigma_{f_2}, \sigma_{f_1} - \sigma_{f_4})$
Cell-based functions:
For $0 \le i, j, k \le p_C - 2$

$$\phi_{(i,j,k)}^C = L_{i+2}(2x - 1) L_{j+2}(2y - 1) L_{k+2}(2z - 1). \tag{A.10}$$

Moreover, the $H(div)$-*conforming shape functions* for variable polynomial order $p = (\{p_{F_m}\}, p_c)$ on the reference hexahedral element are defined by (see, e.g. [93]):
Face-based functions:
For faces F_m, $m = 1, ..., 6$ with local face-vertex ordering $F_m = [f_1, f_2, f_3, f_4]$, $\lambda_F := \sum_{\alpha=0}^{4} \lambda_{f_\alpha}$ and $(\zeta_F, \eta_F) := (\sigma_{f_1} - \sigma_{f_2}, \sigma_{f_1} - \sigma_{f_4})$.
Lowest-order Raviart-Thomas function:

$$\psi = -\nabla \lambda_F \lambda_F \tag{A.11}$$

Higher-order face-based functions (divergence-free):
For $0 \le i, j \le p_{F_m} - 1$

$$\psi_{(i,j)}^{F_m} = curl((\nabla L_{i+2}(\xi_F) L_{j+2}(\eta_F) - L_{i+2}(\xi_F) \nabla L_{j+2}(\eta_F)) \lambda_F), \tag{A.12}$$

$$\psi_{(0,j)}^{F_m} = curl(L_{j+2}(\eta_F) \lambda_F \nabla \xi_F), \tag{A.13}$$

$$\psi_{(i,0)}^{F_m} = curl(L_{i+2}(\xi_F) \lambda_F \xi_F), \tag{A.14}$$

Cell-based functions:
For $0 \le i, j, k \le p_C - 1$ with $(\xi, \eta, \zeta) := (2x-1, 2y-1, 2z-1)$
Type 1: (divergence-free)

$$\begin{aligned}
\psi_{(}i,j,k)^{C,1} &= 4L_{i+1}(\xi)l_k(\eta)l_k(\zeta)\mathbf{e}_x - 4l_i(\xi)l_j(\eta)L_{k+2}(\zeta)\mathbf{e}_z, && \text{(A.15)}\\
\psi_{(}i,j,k)^{C,1} &= 4l_i(\xi)L_{j+2}(\eta)l_k(\zeta)\mathbf{e}_y - 4l_i(\xi)l_j(\eta)L_{k+2}(\zeta)\mathbf{e}_z, && \text{(A.16)}\\
\psi_{(}0,j,k)^{C,1} &= 2L_{j+2}(\eta)l_{k+1}(\zeta)\mathbf{e}_y - 4l_{j+1}(\eta)L_{k+2}(\zeta)\mathbf{e}_z, && \text{(A.17)}\\
\psi_{(}i,0,k)^{C,1} &= 2l_{i+1}(\xi)L_{k+2}(\zeta)\mathbf{e}_z - 2L_{i+2}(\xi)l_{k+1}(\zeta)\mathbf{e}_x, && \text{(A.18)}\\
\psi_{(}i,j,k)^{C,1} &= 2L_{i+2}(\xi)l_{j+1}(\eta)\mathbf{e}_x - 2l_{i+1}(\xi)L_{j+2}(\eta)\mathbf{e}_y, && \text{(A.19)}
\end{aligned}$$

Type 2:

$$\begin{aligned}
\psi^{C,2}_{(i,j,k)} &= L_{i+2}(\xi)l_j(\eta)l_k(\zeta)\mathbf{e}_x + l_i(\xi)L_{j+2}(\eta)l_k(\zeta)\mathbf{e}_y, && \text{(A.20)}\\
\psi^{C,2}_{(0,j,k)} &= L_{j+2}(\eta)l_k(\zeta)\mathbf{e}_y + l_{j+1}(\eta)L_{k+2}(\zeta)\mathbf{e}_z, && \text{(A.21)}\\
\psi^{C,2}_{(i,0,k)} &= l_{i+1}(\xi)L_{k+2}(\zeta)\mathbf{e}_z + L_{i+2}(\xi)l_{k+1}(\zeta)\mathbf{e}_x, && \text{(A.22)}\\
\psi^{C,2}_{(i,j,0)} &= L_{i+2}(\xi)l_{j+1}(\eta)\mathbf{e}_x + l_{i+1}(\xi)L_{j+2}(\eta)\mathbf{e}_y, && \text{(A.23)}
\end{aligned}$$

Type 3:

$$\begin{aligned}
\psi^{C,3}_{(i,0,0)} &= L_{i+2}(\xi)\mathbf{e}_x, && \text{(A.24)}\\
\psi^{C,3}_{(0,j,0)} &= L_{j+2}(\eta)\mathbf{e}_y, && \text{(A.25)}\\
\psi^{C,3}_{(0,0,k)} &= L_{k+2}(\zeta)\mathbf{e}_z. && \text{(A.26)}
\end{aligned}$$

A more detailed description of the shape functions can be found in [93].

Acronyms and symbols

Acronyms

3D	Three dimensional
BCDG	Boundary conformal Discontinuous Galerkin method
CAD	Computer-aided design
DG	Discontinuous Galerkin method
DOF	Degrees of freedom
EBCDG	Extension of the boundary conformal Discontinuous Galerkin method
ES	Electrostatic approximation of the Maxwell's equations
EQS	Electro-quasistatic approximation of the Maxwell's equations
LDG	Local Discontinuous Galerkin method
FD	Finite Difference scheme
FDTD	Finite Difference Time Domain method
FE	Finite Element method
FIT	Finite Integration Technique
FV	Finite Volume method
MQS	Magneto-quasistatic approximation of the Maxwell's equations
MS	Magnetostatic approximation of the Maxwell's equations
PFC	Partially Filled Cell approach

General Symbols and Conventions

Notation	**Description**
$\mathbb{R}$	Real numbers
$\mathbb{N}$	Natural numbers
$\mathbf{x}, \mathbf{x}^T$	Column vector and row vector
$\mathbf{x} \cdot \mathbf{y}$	Dot product
$\mathbf{x} \times \mathbf{y}$	Cross product
$\mathbf{A}$	Matrix
$\nabla \cdot \mathbf{F}$	Divergence of a vector field $\mathbf{F}$
$\nabla \times \mathbf{F}$	Rotation of a vector field $\mathbf{F}$
$\{u\}$	Jump of a scalar function u
$[\![u]\!]$	Average of a scalar function u
$\{\mathbf{q}\}$	Jump of a vector function $\mathbf{q}$
$[\![\mathbf{q}]\!]$	Average of a vector function $\mathbf{q}$

Greek letters

Notation	Description
κ	Electric conductivity (SI unit: S/m)
σ	Surface charge density (SI unit: C/m^2)
ρ	Electric charge density (SI unit: $C/m^3 = As/m^3$)
ϵ_0	Permittivity of vacuum (SI unit: $As/Vm = F/m$)
μ_0	Permeability of vacuum (SI unit: Vs/Am)
χ_e	Dielectric susceptibility
χ_m	Magnetic susceptibility
ϵ	Permittivity of a material
μ	Permeability of a material
ϵ_r	Relative permittivity of a material
μ_r	Relative permeability of a material
τ_e	Electro-quasistatic charge relaxation time
τ_m	Magneto-quasistatic diffusion time
τ_{em}	Electromagnetic wave transit time
τ	Characteristic time of an excitation
ϕ	Electric potential function
ϕ_h	Approximate solution of the electric potential
$\tilde{\phi}_h$	Numerical flux term of the electric potential
$\boldsymbol{\phi}$	Vector containing potential DOF
ω	Angular frequency
Ω	Computational domain
Ω_h, Ω_h^C	Mesh or set of elements
Ω_h^N	Set of normal elements
Ω_h^{CC}	Set of cut-cells
$\Gamma = \partial\Omega$	Boundary of the computational domain
Γ_D	Boundary with Dirichlet boundary conditions
Γ_N	Boundary with Neumann boundary conditions
Γ_{Int}	Interior material interfaces
υ_i	Scalar test function
$\boldsymbol{\tau}_i$	Vector test function
Σ_h	Finite element space
$\varepsilon_{L^2}^{rel}$	Relative error measured in the L^2 norm

Roman letters

Notation	Description
E	Electric field strength (SI unit: V/m)
H	Magnetic field strength (SI unit: A/m)
D	Electric displacement flux density (SI unit: $C/m^2 = As/m^2$)
B	Magnetic flux density (SI unit: $T = Vs/m^2$)
J	Electric current density (SI unit: A/m^2)
P	Polarization density
M	Magnetization density

$\mathbf{J}_l$	Conduction current density (SI unit: A/m^2)
$\mathbf{J}_s$	Source current density (SI unit: A/m^2)
$\mathbf{J}_c$	Convection current density (SI unit: A/m^2)
$\mathbf{v}$	Velocity (SI unit: m/s)
$\mathbf{J}_F$	Surface current density (SI unit: A/m^2)
$\mathbf{x}$	Spatial parameter
t	Time
A	Particular surface in $\mathbb{R}^3$
V	Particular volume in $\mathbb{R}^3$
c	Speed of light (SI unit: m/S)
l	Size length
K_i, K_i^C	Element of the mesh
$\hat{K}$	Reference element
N	Number of elements
$\mathcal{N}$	Set of nodes
$\mathcal{F}$	Set of faces
$\mathcal{E}$	Set of edges
$\mathcal{F}_N$	Set of boundary faces with Neumann boundary conditions
$\mathcal{F}_D$	Set of boundary faces with Dirichlet boundary conditions
$\mathcal{F}_0$	Set of interior faces
n_i	A particular node of the mesh
f_i	A particular face of the mesh
e_i	A particular edge of the mesh
h	Minimum element length
V_h	Finite element space
$\mathcal{S}$	Local finite element space
$\mathcal{U}$	Local finite element space
$\mathcal{P}^p$	Polynomial space of order at most $p \in \mathbb{N}$
$\mathbf{n}$	Unit outward normal vector
$\tilde{\mathbf{D}}_h$	Numerical flux term of the flux density $\mathbf{D}$
$\mathbf{L}$	Operator matrix
$\mathbf{W}$	Operator matrix
$\mathbf{M}$	Mass matrix
$\mathbf{G}$	Gradient matrix
$\mathbf{S}$	Stability term matrix
$\mathbf{f}, \mathbf{g}$	Vector containing boundary conditions
$\mathbf{d}$	Vector containing flux density DOF
$\mathbf{C}_{12}$,C_{11}	Parameters of the LDG flux formulation

Bibliography

[1] (2012) CST STUDIO SUITE 2012, CST Computer Simulation Technology. [Online]. Available: http://www.cst.com/ (Cited on pages 4, 74, 83, and 100.)

[2] (2012) Open CASCADE Technology, 3D modeling and numerical simulation. [Online]. Available: http://www.opencascade.org/ (Cited on pages 44, 45, 46, 48, 50, 51, and 91.)

[3] (2012) PETSc: Portable, Extensible Toolkit for Scientific Computation. [Online]. Available: http://www.mcs.anl.gov/petsc/ (Cited on pages 63 and 92.)

[4] G. Antonini and A. Ruehli, "Fast multipole and multifunction PEEC methods," *IEEE Trans. Mobile Comput.*, vol. 2, no. 4, pp. 288–298, 2003. (Cited on page 10.)

[5] J. H. Argyris and S. Kelsey, "Energy Theorems and Structural Analysis," *Aircraft Engineering and Aerospace Technology*, vol. 26, pp. 410 – 422, 1954. (Cited on page 19.)

[6] D. Arnold, "An interior penalty finite element method with discontinuous elements," *SIAM J. Numer. Anal.*, vol. 19, pp. 742–760, 1982. (Cited on page 37.)

[7] D. Arnold, F. Brezzi, B. Cockburn, and D. Marini, "Unified Analysis of Dicontinuous Galerkin Methods for Elliptic Problems," *SIAM J. Numer. Anal*, vol. 39, pp. 1749–1779, 2002. (Cited on pages 22, 23, 24, 27, 37, and 68.)

[8] F. Bassi and R. Rebay, "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations," *J. Comput. Phys.*, vol. 131, pp. 267–279, 1997. (Cited on page 20.)

[9] P. Bastian and C. Engwer, "An Unfitted Finite Element Method using Dicontinuous Galerkin," *Int. J. Numer. Meth. Engng.*, vol. 3, pp. 1–20, 2008. (Cited on pages 3, 37, 47, and 80.)

[10] C. Baumann and J. T. Oden, "A Discontinuous hp finite element method for the Euler and Navier-Stokes equations," *J. Numer. Methods Fluids*, vol. 31, pp. 79–95, 1999. (Cited on page 20.)

[11] G. Benderskaya, "Numerical Methods for Transient Field-Circuit Coupled Simulations based on the Finite Integration Technique and a Mixed Circuit Formulation," Ph.D. dissertation, TU Darmstadt, 2007. (Cited on page 58.)

[12] D. Braess, *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer-Verlag Berlin Heidelberg, 2007. (Cited on pages 19 and 20.)

[13] S. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, 2002. (Cited on pages 19, 20, and 21.)

[14] I. Bronstein, K. Semendjajew, G. Musiol, and H. Mühlig, *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2001. (Cited on page 91.)

[15] E. Burman and P. Hansbo, “Fictitious domain finite element methods using cut elements: I. a stabilized lagrange multiplier method,” *Comput. Methods Appl. Mech. Engrg.*, vol. 199, pp. 2680–2686, 2010. (Cited on page 19.)

[16] ——, “Fictitious domain finite element methods using cut elements: II A stabilized Nitsche method,” *Applied Numerical Mathematics*, vol. 62, pp. 328–341, 2012. (Cited on pages 1, 37, 38, and 44.)

[17] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau, “An a priori error analysis of the local discontinuous Galerkin method for elliptic problems,” *SIAM J. Numer. Anal.*, vol. 38, pp. 1676–1706, 2000. (Cited on pages 24, 26, 27, 30, and 80.)

[18] B. Cockburn, “Discontinuous Galerkin methods,” *Z. Angew. Math. Mech.*, vol. 83, pp. 731–754, 2003. (Cited on pages 20, 23, 27, 29, and 30.)

[19] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, “Unified hybridization of discontinuous galerkin, mixed, and continuous galerkin methods for second order elliptic problems,” *SIAM J. Numer. Anal*, vol. 47(2), pp. 1319–1365, 2009. (Cited on page 38.)

[20] B. Cockburn and C. Shu, “The local discontinuous Galerkin method for time-dependent convection-diffusion systems,” *SIAM J. Numer. Anal.*, vol. 35, pp. 2440–2463, 1998. (Cited on pages 18, 20, 21, 22, 23, 27, 30, and 80.)

[21] G. Cohen, X. Ferrieres, and S. Pernet, “A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell’s equations in time domain,” *Journal of Computational Physics*, vol. 217, pp. 340–363, 2006. (Cited on page 20.)

[22] R. Courant, “Variational Methods for the Solution of problems of equilibrium and vibrations,” *Bull. Amer. Math. Soc*, 1943. (Cited on page 19.)

[23] S. Dey and R. Mittra, “A locally conformal finite-difference time-domain (FDTD) algorithm modeling modeling three-dimensional perfectly conducting objects,” *IEEE Microwave Guided Wave Lett.*, vol. Vol. 7, pp. pp.273–275., 1997. (Cited on pages 1, 3, 19, and 36.)

[24] H. Dirks, “Quasi-stationary fields for microelectronic applications,” *Electrical Engineering*, vol. 79, pp. 145–155, 1996. (Cited on page 10.)

[25] A. Düster, J. Parvizian, Z. Yang, and E. Rank, “The finite cell method for three-dimensional problems of solid mechanics,” *Comput. Methods Appl. Mech. Engrg.*, vol. 197, pp. 3768–3782, 2008. (Cited on page 50.)

[26] B. Engquist, A. Tornberg, and R. Tsai, “Discretization of Dirac delta functions in level set methods,” *Journal of Computational Physics*, vol. 207, pp. 28–51, 2004. (Cited on page 44.)

[27] K. Eriksson, C. Johnson, and V. Thomée, “Time discretization of parabolic problems by the discontinuous galerkin method,” *RAIRO Model. Math. Anal. Numér.*, vol. 19, pp. 611–643, 1985. (Cited on page 20.)

[28] C. Farhat, I. Harari, and L. Franca, “The discontinuous enrichment method,” *Comput. Methods Appl. Mech. Engrg.*, vol. 190, pp. 6455–6479, 2001. (Cited on page 38.)

[29] K. Fidkowski and D. Darmofal, “A triangular cut-cell adaptive method for higher order discretizations of the compressible Navier-Stokes equations,” *Journal of Computational Physics*, vol. 225, pp. 1653–1672, 2007. (Cited on pages 1, 3, and 37.)

[30] A. Froehlcke, E. Gjonaj, and T. Weiland, “A boundary conformal Discontinous Galerkin approach for electro-quasistatic problems,” *Scientific Computing in Electrical Engineering (SCEE 2010), The European Consortium for Mathematics in Industry, Springer-Verlag, Berlin/Heidelberg*, vol. ISBN: 978-3-642-22452-2, 2012. (Cited on page 3.)

[31] ——, “A boundary conformal Discontinuous Galerkin approach for transient electro-quasistatic problems with moving boundaries,” *International Conference on Electromagnetics in Advanced Applications (ICEAA 2012),Cape Town, WP, South Africa,*, vol. ISBN: 978-1-4673-0333-0, pp. 606–609, 2012. (Cited on page 3.)

[32] ——, “A boundary conformal Discontinuous Galerkin Approach for electro-quasistatic field problems on Cartesian grids,” *International Journal of Computational Science and Engineering*, forthcoming. (Cited on pages 3 and 38.)

[33] E. Gjonaj and T. Weiland, “A projection penalization approach for the high-order DG-FEM in the time domain,” *RADIO SCIENCE*, vol. Vol. 46, RS0E10, p. 10 PP., 2011. (Cited on pages 20 and 55.)

[34] R. Gracie, H. Wang, and T. Belytschko, “Blending in the extended finite element method by discontinuous Galerkin and assumed strain methods,” *Int. J. Numer. Meth. Engng.*, vol. DOI: 10.1002, 2007. (Cited on page 31.)

[35] A. Hansbo and P. Hansbo, “An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems,” *Comput. Methods Appl. Mech. Engrg.*, vol. 191(47-48), pp. 5537–5552, 2002. (Cited on page 37.)

[36] A. Hansbo, P. Hansbo, and M. Larson, “A finite element method on composite grids based on Nitsche’s method,” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 37, pp. 495–514, 2003. (Cited on page 37.)

[37] H. Haus and J. Melcher, *Electromagnetic Fields and Energy*. Prentice-Hall. Inc., 1989. (Cited on pages 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, and 79.)

[38] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis and Applications*. Springer, 2008. (Cited on pages 1, 20, 21, 23, 26, 31, 51, and 77.)

[39] J. Hesthaven and T. Warburton, “High-order nodal discontinuous Galerkin methods for the Maxwell eigenvalue problem,” *Phil. Trans. R. Soc. Lond. A*, vol. 362, pp. 493–524, 2004. (Cited on page 20.)

[40] R. Hiptmair and C. Schwab, *Numerical treatment of partial differential equations*. Lecture Notes, Dep. Mathematics, ETH Zurich, 2004. (Cited on page 21.)

[41] S. Hofmann, M. Mitrea, and M. Taylor, “Geometric and Transformational Properties of Lipschitz Domains, Semmes-Kenig-Toro Domains, and Other Classes of Finite Perimeter Domains,” *The Journal of Geometric Analysis*, vol. Vol. 17, No. 4,, pp. 593–647, 2007. (Cited on page 21.)

[42] T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, INC. Mineola, New York, 2000. (Cited on page 21.)

[43] T. Hughes, J. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Comput. Methods Appl. Mech. Engrg.*, vol. 194, p. 4135Ű4195, 2005. (Cited on page 44.)

[44] J. Jackson, *Classical Electrodynamics*. John Wiley & Sons, Inc., 1999. (Cited on pages 5 and 8.)

[45] P. Jacobsson and T. Rylander, "A Finite Element Method Approach to the Finite-Difference Time-Domain Scheme with Cut Cells," *Electromagnetics*, vol. 30, pp. 82–93, 2010. (Cited on pages 3 and 37.)

[46] P. Jamet, "Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain," *SIAM J. Numer. Anal*, vol. 15, pp. 912–928, 1978. (Cited on page 20.)

[47] K. Jänich, *Vektoranalysis*. Springer-Verlag, 2005. (Cited on page 49.)

[48] A. Johansson and M. Larson, "A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary," *Numer. Math DOI 10.2007/s00211-012-0497-1*, 2012. (Cited on pages 1, 3, 37, 38, 44, 47, 50, and 80.)

[49] S. Keim, "Optische Diagnose an singulären Tropfen auf polymeren Isolierstoffoberflächen im elektrischen Feld," Ph.D. dissertation, TU Darmstadt, 2003. (Cited on pages 1, 15, and 16.)

[50] S. Koch, "Quasistatische Feldsimulation auf der Basis von Finiten Elementen und Spektralmethoden in der Anwendung auf supraleitende Magnete," Ph.D. dissertation, TU Darmstadt, 2009. (Cited on pages 8 and 10.)

[51] P. Lax and R. Richtmyer, "Survey of the stability of linear finite difference equations," *Comm. Pure Appl. Math.*, vol. 9, pp. 267–293, 1956. (Cited on page 30.)

[52] Y. Lee and A. Requicha, "Algorithms for Computing the Volume and Other Integral Properties of Solids. I. Known Methods and Open Issues," *Communications of the ACM*, vol. 25(9), pp. 635–641, 1982. (Cited on pages 48 and 49.)

[53] G. Lehner, *Elektromagnetische Feldtheorie*. Springer Verlag Berlin Heidelberg, 2006. (Cited on pages 7 and 17.)

[54] X. Li and J. Demmel, "SuperLU_DIST: A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems," *ACM Transactions on Mathematical Software*, vol. 29, No. 2, pp. 110–139, 2003. (Cited on page 63.)

[55] J. Macey, *Ray-tracing and other Rendering Approaches*. Lecture Notes, MSc Computer Animation and Visual Effects, University of Bournemouth. (Cited on page 46.)

[56] C. Makridakis and I. Babuska, "On the stability of the discontinuous Galerkin method for the heat equation," *SIAM J. Numer. Anal.*, vol. 34, pp. 389–401, 1997. (Cited on page 20.)

[57] A. Massing, "Analysis and Implementation of Finite Element Methods on Overlapping and Fictitious Domains," Ph.D. dissertation, Departement of Informatics, Faculty of Mathematics and Natural Science, University of Oslo, 2012. (Cited on pages 3, 19, 37, 38, 50, and 80.)

[58] J. C. Maxwell, "A Treatise on Electricity and Magnetism," *Dover Publications Inc.*, vol. ISBN 0-486-60637-6, 1954. (Cited on page 5.)

[59] B. Müller, F. Kummer, M. Oberlack, and Y. Wang, "Simple multidimensional integration of discontinuous functions with application to level set methods," *Int. J. Numer. Meth. Engng.*, vol. 92, pp. 637–651, 2012. (Cited on page 44.)

[60] P. Monk, *Finite Element Methods for Maxwell's Equations*. Oxford Science Publications, 2003. (Cited on pages 1, 19, 21, 33, and 35.)

[61] P. J. Nahin, *Oliver Heaviside: The Life, Work and Times of an Electrical Genius of the Victorian Age*. Johns Hopkins Univ Pr, 2002. (Cited on page 5.)

[62] J. Nitsche, "Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind." *Abh. Math. Sem. Univ. Hamburg*, vol. 36, pp. 9–15, 1971. (Cited on page 37.)

[63] W. L. Oberkampf, T. Trucano, and C. Hirsch, "Verification, Validation, and Predictive Capability in Compuational Engineering and Physics," *Proceedings of "Foundation for Verification and Validation in the 21st Century Workshop", John Hopkins University, Maryland*, pp. 1–74, 2002. (Cited on page 74.)

[64] S. Osher and R. P. Fedkiw, "Level Set Methods: An Overview and Some Recent Results," *Journal of Computational Physics*, vol. 169, pp. 463–502, 2001. (Cited on page 44.)

[65] I. Perugia and D. Schötzau, "On the Coupling of Local Discontinuous Galerkin and Conforming Finite Element Methods," *Journal of Scientific Computing*, vol. 16 (4), pp. 411–433, 2001. (Cited on pages 38, 57, and 80.)

[66] J. Purvis and J. Burkhalter, "Prediction of critical Mach number for store configurations," *AIAA Journal*, vol. 17 (11), pp. 1170–1177, 1979. (Cited on pages 3 and 36.)

[67] W. Reed and T. Hill, "Triangular mesh methods for the neutron transport equation," *Los Alamos Scientific Laboratory Report*, pp. LA–UR–73–479, 1973. (Cited on page 20.)

[68] J. Schöberl. NGSolve - 3D Finite Element Solver. [Online]. Available: http://www.hpfem.jku.at/ngsolve/ (Cited on page 50.)

[69] M. Schäfer, *Numerik im Maschinenbau*. Springer Verlag, 1999. (Cited on page 59.)

[70] P. Scholz, "Analysis and Numerical Modeling of Inductively Coupled Antenna Systems," Ph.D. dissertation, TU Darmstadt, 2010. (Cited on pages 8 and 10.)

[71] D. Schötzau, C. Schwab, and A. Toselli, "Mixed hp-dgfem for incompressible flows," *SIAM J. Numer. Anal*, vol. 40, No. 6, pp. 2171–2194, 2003. (Cited on page 68.)

[72] P. Silvester and M. Chari, "Finite Element Solution of Saturable Magnetic Field Problems," *IEEE Trans. on PAS*, vol. 89, pp. 1642–1651, 1970. (Cited on page 19.)

[73] B. Simeon, *Numerik der gewöhnlicher Differentialgleichungen, WS 03/04*. Lecture Notes, TU München, 2004. (Cited on pages 60 and 61.)

[74] ——, *Numerik der partiellen Differentialgleichungen, SS 04*. Lecture Notes, TU München, 2004. (Cited on page 63.)

[75] H. Songoro, E. Gjonaj, and T. Weiland, "Computational modeling of water droplet deformations in strong electric fields," *International Conference on Electromagnetics in Advanced Applications - ICEAA 2012, Cape Town, South Africa.*, vol. 10.1109/ICEAA.2012.6328644, pp. 333 – 336, 2012. (Cited on pages 19, 39, and 64.)

[76] H. Spachmann, "Die Methode der Finiten Integration höherer Ordnung zur numerischen Berechnung elektromagnetischer Felder," Ph.D. dissertation, TU Darmstadt, 2003. (Cited on page 19.)

[77] J. Stoer, *Einführung in die Numerische Mathematik 1*. Springer-Verlag Berlin Heidelberg New York, 2004. (Cited on pages 59, 86, and 87.)

[78] J. Stoer and R. Bulirsch, *Einführung in die Numerische Mathematik II*. Springer-Verlag Berlin Heidelberg New York, 1978. (Cited on pages 58, 59, 60, 61, and 62.)

[79] M. Sussman and E. G. Puckett, "A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows," *Journal of Computational Physics*, vol. 162, p. 301Ű337, 2000. (Cited on page 44.)

[80] P. Thoma, "Zur numerischen Lösung der Maxwellschen Gleichungen im Zeitbereich," Ph.D. dissertation, TU Darmstadt, 1997. (Cited on pages 1, 3, 19, and 36.)

[81] A. Tornberg, "Multi-dimensional quadrature of singular and discontinuous functions," *BIT Numerical Mathematics*, vol. 42, pp. 644–669, 2002. (Cited on page 44.)

[82] G. Turnbull. IEEE Global History Network - STARS: Maxwell's Equations. [Online]. Available: http://www.ieeeghn.org/wiki/index.php/STARS:Maxwell's_Equations (Cited on pages 5 and 9.)

[83] M. Turner, R. Clough, H. Martin, and L. Topp, "Stiffness and Deflection Analysis of Complex Structures," *J. Aero. Sci.*, vol. 23, pp. 805–823, 1956. (Cited on page 19.)

[84] S. Ulbrich, *Partial Differential Equations, WS 03/04*. Lecture Notes, TU München, 2004. (Cited on page 11.)

[85] U. van Rienen, *Numerical Methods in Computational Electrodynamics, Linear Systems and Practical Applications*. Springer-Verlag Berlin Heidelberg New York, 2001. (Cited on pages 8, 17, 18, 19, 20, and 63.)

[86] A. Vande, P. Saucez, and W. Schiesser, *Adaptive Method of Lines*. Chapman &Hall/CRC, 2001. (Cited on pages 18 and 58.)

[87] G. Ventura, "On the elimination of quadrature subcells for dicontinuous functions in the eXtended Finite Element method," *Int. J. Numer. Meth. Engng.*, vol. 66, pp. 761–795, 2006. (Cited on page 44.)

[88] T. Weiland, *Verfahren und Anwendungen der Feldsimulation I, Skriptum zur Vorlesung, TU Darmstadt*. (Cited on page 20.)

[89] ——, "Time domain electromagnetic field computation with finite difference methods," *International Journal of Numerical Modelling: Electronic Networks,Devices and Fields*, vol. 9, pp. 295–319, 1996. (Cited on pages 18 and 20.)

[90] ——, *Skriptum zu Grundlagen der Elektrodynamik und Technischen Elektrodynamik*. Lecture Notes WS11/12, Institut für Theorie Elektromagnetischer Felder, 2012. (Cited on pages 6, 8, 17, 74, and 76.)

[91] ——, "A discretization model for the solution of Maxwell's equations for six-component fields," *AEÜ*, vol. 1977, pp. 116–120, 31. (Cited on pages 1 and 18.)

[92] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, pp. 302–307, 1966. (Cited on pages 1, 18, and 20.)

[93] S. Zaglmayr, "High order finite element methods for electromagnetic field computation," Ph.D. dissertation, Johannes Kepler Universität Linz, 2006. (Cited on pages 21, 31, 32, 33, 34, 50, 52, 102, 103, and 104.)

[94] Z. Zhu, B. Song, and J.K.White, "Algorithm in fastimp: A fast and wideband impedance extraction program for complicated 3-d geometries," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 7, pp. 981–998, 2005. (Cited on page 10.)

[95] P. Zunino, L. Cattaneo, and C. Colciago, "An unfitted interface penalty method for the numerical approximation of contrast problems," *Applied Numerical Mathematics*, vol. 61, pp. 1059–1076, 2011. (Cited on pages 37 and 38.)

Danksagung

An dieser Stelle möchte ich mich bei allen herzlich bedanken, die in den letzten Jahren zum Gelingen dieser Arbeit beigetragen haben. Insbesondere gilt mein Dank:

- Prof. Dr.-Ing. Thomas Weiland für die wissenschaftliche Betreuung und die ausgezeichneten Arbeitsbedingungen am Institut.
- PD Dr. rer. nat. habil. Erion Gjonaj für die hervorragende fachliche Betreuung.
- Prof. Dr. rer. nat. Michael Schäfer für die freundliche Übernahme des Korreferats.
- Ulrich Römer, Jens Trommler, Kynthia Stavrakakis, Harald Songoro und Thomas Lau für die vielen wertvollen fachlichen Ratschläge.
- Heike Seiler, Dragos Munteanu, Marianne Dorn und Achim Veuskens für die administrative und technische Unterstützung in den letzen Jahren.
- allen meinen Kolleginnen und Kollegen am Institut für Theorie Elektromagnetischer Felder der TU-Darmstadt und an der Graduate School of Computational Engineering für die angenehme gemeinsame Zeit.
- der Graduate School of Computational Engineering für die finanzielle Unterstützung in den letzten drei Jahren.
- meinen Eltern für die bedingungslose Unterstützung in jeder Hinsicht.

Ganz besonders möchte ich meinem Freund Michael danken für die großartige Unterstützung in den letzten Jahren und dafür, dass er immer für mich da ist. Danke!

Curriculum Vitae

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten.